ALGORITHMS FOR VLSI STEREO VISION CIRCUITS APPLIED TO AUTONOMOUS ROBOTS

THÈSE N^{O} 2058 (1999)

PRÉSENTÉE AU DÉPARTEMENT D'INFORMATIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Yuri LOPEZ DE MENESES NOVOSILZOV

M.Sc. in Electrical Engineering, University of Southern California, Los Angeles, Etats-Unis

de nationalité espagnole

acceptée sur proposition du jury:

Prof. J.-D. Nicoud, directeur de thèse
Dr X. Arreguit, rapporteur
Dr K. Konolige, rapporteur
Prof. R. Siegwart, rapporteur
Prof. E. Vittoz, rapporteur

Lausanne, EPFL
1999

Abstract

Since the inception of Robotics, visual information has been incorporated in order to allow the robots to perform tasks that require an interaction with their environment, particularly when it is a changing environment. Depth perception is a most useful information for a mobile robot to navigate in its environment and interact with its surroundings. Among the different methods capable of measuring the distance to the objects in the scene, stereo vision is the most advantageous for a small, mobile robot with limited energy and computational power.

Stereoscopy implies a low power consumption because it uses passive sensors and it does not require the robot to move. Furthermore, it is more robust, because it does not require a complex optic system with moving elements. On the other hand, stereo vision is computationally intensive. Objects in the scene have to be detected and matched across images.

Biological sensory systems are based on simple computational elements that process information in parallel and communicate among them. Analog VLSI chips are an ideal substrate to mimic the massive parallelism and collective computation present in biological nervous systems. For mobile robotics they have the added advantage of low power consumption and high computational power, thus freeing the CPU for other tasks.

This dissertation discusses two stereoscopic methods that are based on simple, parallel calculations requiring communication only among neighboring processing units (local communication). Algorithms with these properties are easy to implement in analog VLSI and they are also very convenient for digital systems.

The first algorithm is phase-based. Disparity, i.e., the spatial shift between left and right images, is recovered as a phase shift in the spatial-frequency domain. Gábor functions are used to recover the frequency spectrum of the image because of their optimum joint spatial and spatial-frequency properties.

The Gábor-based algorithm is discussed and tested on a Khepera miniature mobile robot. Two further approximations are introduced to ease the analog VLSI and digital implementations.

The second stereoscopic algorithm is difference-based. Disparity is recovered by a simple calculation using the image differences and their spatial derivatives. The algorithm is simulated on a digital system and an analog VLSI implementation is proposed and discussed.

The thesis concludes with the description of some tools used in this research project. A stereo vision system has been developed for the Webots mobile robotics simulator, to simplify the testing of different stereo algorithms. Similarly, two stereo vision turrets have been built for the Khepera robot.

Version Abrégée

Depuis les origines de la robotique, l'information visuelle a été incorporée pour permettre aux robots d'interagir avec leur environnement, en particulier lors qu'il s'agit d'un environnement changeant. La perception de la profondeur, i.e., de la distance, est une information très utile pour un robot qui doit naviguer dans son environnement et interagir avec son milieu. Parmi les différentes méthodes pour mesurer la distance aux objets d'une scène, la vision stéréoscopique est la plus avantageuse pour un petit robot mobile avec une énergie et une puissance de calcul limitées.

La stéréoscopie implique une moindre consommation de puissance, parce qu'elle utilise des capteurs passifs et qu'elle n'exige pas que le robot soit en mouvement. Qui plus est, elle est plus robuste, car elle ne nécessite pas d'une optique complexe avec des éléments mobiles. En revanche, la vision stéréoscopique est très gourmande en puissance de calcul puisqu'il faut détecter les objets dans la scène et les apparier avec ses correspondants sur l'autre image.

Les systèmes sensoriels biologiques se basent sur des éléments de calcul simples, qui traîtent l'information en parallèle et communiquent entre eux. Les chips de VLSI analogique sont un substrat idéale pour recréer le parallélisme massif et le calcul collectif présent dans les systèmes nerveux biologiques. Pour la robotique mobile ils ont l'avantage ajouté d'une faible consommation de puissance et une grande puissance de calcul, libérant ainsi le processeur central pour d'autres tâches.

Cette thèse présente deux méthodes de vision stéréoscopique qui se basent sur des calculs simples et parallèles qui nécessitent uniquement de la communication entre unités de calcul voisines. Des algorithmes avec des telles propriétés sont plus facilement mis en œuvre en VLSI analogique et ils sont aussi très avantageux pour les systèmes digitaux.

Le premier algorithme est basé sur la phase. La disparité, c'est à dire, le décalage spatial entre les images gauche et droite, est extraite comme un décalage de phase dans le domaine des fréquences spatiales. Des fonctions de Gábor sont employées pour obtenir le spectre de fréquences des images, en raison de leurs propriétés optimales à la fois dans le domaine spatial et des fréquences spatiales.

L'algorithme de Gábor est présenté et vérifié sur un petit robot mobile Khepera. Deux approximations qui visent à faciliter la mise en œuvre en systèmes analogiques et digitaux sont proposées.

Le second algorithme stéréoscopique est basé sur des différences. La disparité est récupérée par un calcul simple qui utilise la différence des images et leur dérivée. L'algorithme est simulé sur un système digital et une mise en œuvre en VLSI analogique est proposée et décrite.

La thèse se termine avec la description de quelques outils employés dans ce projet de recherche. Un système de vision stéréo a été développé pour le simulateur de robots mobiles Webots. Il simplifie la vérification des algorithmes stéréoscopiques. Pareillement, deux tourelles de vision stéréoscopique on été développées pour le robot Khepera.

To Anna and Jelena

Να εύχεσαι νάναι μακρύς ο δρόμος
γεμάτος περιπέτειες, γεμάτος γνώσεις.

Κ. Καβάφη

Acknowledgements

I would like to thank Prof. Jean-Daniel Nicoud for having accepted me into the LAMI and allowed me to pursue this research. Thanks to his energy and team-building skills, the LAMI has been a very exciting place to work in the field of mobile robotics and embedded systems.

Among the people with whom I have shared successes and failures during these last three years, my warmest thanks go to Olivier Carmona, for the innumerable discussions on bioinspired vision, robotics, UNIX and cuisine; to Philip Mächler, engineer extraordinaire, always available to help with any hardware, software or Germanware problem; to Alcherio Martinoli, for imposing discipline on the Kheperas, the assistants and my thoughts; to Jelena Godjevac, for her encouragement, insistence, and just for being "atrocious".

I wish to acknowledge the names of other colleagues that have made possible this research by their help and their knowledge: André Guignard, Edo Franzi, Dario Floreano, Marie-Jo Pellaud, René Beuchat, Olivier Michel, Olivier Matthey, Francesco Mondada, Laurent Tettoni, Eric Fragnière, Michel Pahud, Jean-Marc Koller and Prof. Wulfram Gerstner.

I am particularly indebted to the Centre Suisse d'Electronique et de Microtechnique (CSEM), in the persons of Prof. Eric Vittoz, Olivier Landolt and Xavier Arreguit for their financial support and for introducing me into the amazing world of analog VLSI.

My eternal appreciation goes to Jelena Godjevac for starting the machine of this dissertation, and to Anna Biamonte for keeping the engine well oiled, finely tuned and working for nine months and, literally, up to the last minute.

I am very grateful to those that have kindly accepted to review and discuss some parts or the whole manuscript: Jelena Godjevac, Philip Mächler, Prof. Huber, Gilles Thonet, Eric Fragnière, Prof. Patrick Thiran, Daniel Rodellar and Jürgen Ehrensberger.

I would like to thank the members of the jury—Prof. Petitpierre, Dr. Arreguit, Dr. Konolige, Prof. Siegwart and Prof. Vittoz— for their patience in reading this document and their helpful comments. I appreciate their severeness with my mistakes and their leniency towards me.

Ortega y Gasset said, "yo soy yo y mi circunstancia." This dissertation owes a great deal to its circumstance —the friends that have surrounded me, helped me, encouraged me, supported me and advised me: Anna Biamonte, Daniel Rodellar, Mauro Facchini, Gilles Thonet, Alessandro Iocco, Daniele Costantini and the people of the Club Max. I thank them all.

Contents

Chapter 1

Introduction

Autonomous Robotics

In the near future, mobile, autonomous robotics can be expected to find its first mainstream industrial applications. After the first successful applications for exploratory missions such as the Dante II robot on Mt. Spurr, Alaska, and the Sojourner rover on Mars, mobile robots are starting to find commercial applications such as Husqvarna's SolarMower autonomous lawnmowers and Electrolux's robotic vacuum cleaner. Other applications with recent success are entertainment robots, such as the Sony Aibo dog or the LEGO Mindstorms.

Because they are mobile, these robots are increasingly autonomous. First, they must be energetically autonomous, for mobility complicates the conveying of energy to the robot. Some robots, e.g., the Dante II, are tethered, and the power lines can be sent with the tether. In other robots, such as the Sojourner rover, the remoteness requires an energy autonomy. Most robots, however, dispense with tethers for practical reasons: they would end up tangled in the cables or the tethers would jam on other objects in the environment.

The second kind of autonomy that mobile robots increasingly possess is decisional autonomy. The etymological meaning of autonomous is "capable of making its own laws" [Ste95], that is, of taking its own decisions. Remotely operated robots need to take some decisions by themselves, as the transmission delay can put them in jeopardy before the human operator is aware of it. The time delay for the Sojourner rover was between 10 and 20 minutes[1]; if the operator reacts instantly to a danger signal —a crater hole, a rock, a Martian pedestrian— the command to stop will arrive 30 minutes too late. Therefore shorter, reactive sensorimotor loops are implemented to let the robot take its own decisions when its security is at stake.

To operate these short sensorimotor loops, robots not only need enough computational power, but also the adequate sensors. Computational power seems to be readily available in the microprocessor market, but power consumption and volume can compromise the energetic autonomy of the robot. As for the choice of sensors, it depends on the application, of course, but in general, vision sensors are the most adequate for mobile robots.

[1] It depends on the position of Mars with respect to the Earth.

Vision

If the robot is to detect the obstacles in advance or to look for an object, contact sensors —"whiskers", contact switches, pressure meters— are not a satisfying solution. Sensors that provide information of a distant object or area, i.e., remote sensors[2], offer a better solution. They can be divided into active or passive sensors, depending on whether some type of energy —electromagnetic energy or pressure— has to be emitted for the sensor to read the incoming energy. Active sensors include sonar, laser range-finders, laser strips, radar, lidar, and infrared sensors. They are very efficient, and some are very impressive in their accuracy, but, because they are active, they require spending extra energy, an energy a mobile robot may not have. They are also necessarily bulkier, since they consist of an emitter and a sensor properly speaking.

Passive sensors include cameras and microphones. Cameras or vision sensors are the most convenient, because in most settings there is an external light source: the sun, the moon or the fluorescent lamps in an office. Another advantage of vision sensors is that they produce an information that can be interpreted by humans. This simplifies human intervention in the programming of the sensorimotor loop[3] and produces human-readable information. If the ultimate goal is for humans to see what the robot is looking at, it would be advantageous if the robot could navigate using the same camera that remote operators use to view the scene.

Still, most commercial vision sensors, based on *CCD* cameras, require too much power for an energetically autonomous robot, because they require additional circuitry for the image acquisition (a frame grabber) and of course, the subsequent processor to analyze the image. This is not only power-consuming, but bulky too.

Neuromorphic analog VLSI

Small insects, like flies or bees, are perfectly capable of flying, avoiding obstacles and pursuers without possessing a complicated and voluminous vision system. What is their secret? The secret lies in task-specific sensors, integrating sensing and processing as much as possible, and using a specific paradigm called neural computation. Animal sensors are adapted to the tasks that are the most important for a particular species. For instance, fly eyes concentrate on sensing motion, and the fly becomes blind when it stops moving. Processing is embedded in the sensor itself. The human brain does not receive a raw image as it is sensed in the retina, the light-sensitive area in the eye. On the contrary, the retina performs a gain adaptation, filtering, and selects the relevant information —motion, color, edges— to be sent to the brain. Processing on the visual pathway takes place in multiple, simple, computational units that process the nervous signals in a massively parallel way.

In the early 1980s, Professor Carver Mead began to investigate the use of silicon fabrication techniques to recreate these principles of neural circuits. Analog VLSI was found to be the most suitable medium to emulate these circuits, because it emulates the elementary physical phenomena in the nervous system while submitting to similar constraints : wiring costs, finite and continuous magnitudes, etc. Moreover, they were able to integrate, on the same silicon

[2] Sometimes erroneously called "distal sensors".

[3] Although, as [MÄc99] argues, it may also complicate the task by imposing an anthropomorphic image analysis on a machine.

substrate, sensors and processors —just as in biological systems. This new discipline, called *neuromorphic engineering* profits from a fertile interaction between engineers and neurobiologists. The former gain insight into the mechanisms of nervous systems, and are able to apply them to solve engineering problems; the latter obtain a test bed to simulate and verify their models of neural computation.

Over time, several successful circuits have been developed [Tan89], [Mah94] , [Mea89], and a corpus of knowledge has developed, not only for neural modeling, but also for engineering applications. Industrial applications of analog VLSI have appeared in the last five years [Arr96] [Ven97]. These low-power, compact and robust analog VLSI circuits can provide adequate solutions for the sensor needs of autonomous, mobile robots.

1.1 Purpose of this thesis

In the last five years, analog VLSI circuit designers and particularly neuromorphic engineers, have become very interested in mobile, autonomous robotics, as a means of putting their circuit designs into test [Koc94]. Mobile robots offer a platform that mimics some of the motor capabilities of animals and thus yield an opportunity to simulate complete sensorimotor loops. Analog circuit designers, therefore, start in most cases with a solution, the analog sensor, and then find a robotic problem to which it can apply.

The purpose of this thesis is to interest roboticists and particularly mobile-robot researchers in analog VLSI circuits, so that an analog sensor will be developed to solve a previously existing robotic problem. The contributions toward this goal are:

- To discuss the benefits and inconvenients of analog VLSI vision sensors for mobile robotics.
- To introduce roboticists to the structural elements that make up an analog VLSI sensor and illustrate these by their use in some robotics applications.
- To identify and analyze a particular robotic problem that requires vision sensors: 3-D perception by stereoscopy.
- To examine two stereoscopic algorithms that are likely to be implemented in analog VLSI, but that can also benefit simple digital systems.

1.2 Structure of the thesis

The thesis "est omnis divisa in partes tres" [JC87]. Part I tries to place the subject of this thesis against the background of current scientific research. Chapter 2 introduces the reader to analog VLSI circuits; their advantages, inconvenients and basic elements are discussed and their application to mobile robotics illustrated by some examples. Chapter 3 addresses the topic of stereo vision and describes the mechanisms found in Nature and reviews stereoscopy solutions used in Computer Vision.

Part II focuses on two stereoscopic algorithms. Chapter 4 dwells on an algorithm based on the phase of the Gábor functions. The algorithm is developed and tested on a Khepera mobile

robot. Chapter 5 introduces a differential stereo algorithm, which is simpler, and provides experimental data and a proposed analog VLSI implementation.

The smaller part III, blurred in the foreground, describes some of the hardware and software tools that have been developed in this project. Chapter 6 describes a simulated analog retina that has been incorporated to a mobile robotics simulator, and chapter 7 illustrates two vision systems developed for a small mobile robot.

Chapter 8 concludes the main material and draws an overall conclusion on the achievements of this project and the further problems that deserve to be explored.

This thesis finds itself at the confluence of three rivers: mobile robotics, analog electronics and signal processing. Since it is difficult to deftly navigate through the rapids and white water of all three of them, a glossary has been provided at the end of the document, with brief definitions of the most important terminology in the text. The glossary terms are typed in italic face the first time they appear at the beginning of each chapter.

Part I

Background

Chapter 2

On VLSI Vision Sensors for Mobile Robotics

Contents

2.1 Introduction

Mobile, autonomous robots rely strongly on their sensing capabilities to attain their goals. Because of their mobility, they often operate in highly dynamic environments, interacting with other robots or humans and therefore need to sense the changes in their surroundings. Sensors transduce, that is transform, a physical quantity (light, pressure, temperature, etc) into an electrical magnitude (voltage, current or charge) that can be processed by the robot. Depending on their needs, robots are equipped with light sensors or cameras, temperature sensors, pressure or contact sensors, radioactivity sensors, distance sensors (sonar, laser range-finders, infrared sensors), and many others.

Vision sensors are key among the possible sensors used for robots, because of their parallel with human perception and the distal information they provide. They produce a signal that engineers can directly interpret and understand, by comparing it with the image they are seeing themselves. This in turn is perceived as an advantage for the subsequent image analysis. But the main feature of vision sensors is their ability to provide information on distant objects, thus allowing the robot to plan its behavior well in advance.

This chapter discusses a particular type of sensors, analog VLSI vision sensors, within the frame of mobile robotics. The benefits and inconveniences of such sensors are presented, followed by some examples and applications found in literature.

2.2 Benefits from Analog VLSI

VLSI sensors are integrated circuits (ICs) that transduce a physical quantity, such as light, by exploiting the properties of the medium, silicon, and the elementary IC device, the transistor. The current state of technology affords a huge density (transistors/mm^2) [Pig98], and therefore we speak about Very Large Scale Integration (VLSI) processes.

VLSI circuits, as any other circuit, can be discrete or continuous in time and operate with discrete or continuous signals [Sar98], depending on the way the information is processed and stored.

Whereas most of the current VLSI circuits, such as microprocessors and memories, are digital, analog circuits have a niche market, particularly in the field of sensors. In the next subsection I will review the reasons that make it so, focusing on the particular advantages that *analog VLSI* can offer to mobile, autonomous robotics.

2.2.1 Real-time sensing

Physical magnitudes that we are interested in sensing, and among them light intensity, are continuous in time, and indeed they can have a temporal variation (frequency) of several kilohertz. Thus, only a continuous system is able to sense these magnitudes without suffering from artifacts due to aliasing.

2.2.2 Low power consumption

Compared to other electronic systems, and particularly digital VLSI, A-VLSI sensors offer a lower power consumption.

Power consumption in digital circuits

Power consumption in digital systems is modelled by the following equation [Pig98]:

$$P_D = a \cdot f \cdot C \cdot V_{DD}^2 + I_{sc} \cdot f \cdot V_{DD} + I_{leak} \cdot V_{DD} \tag{2.1}$$

The first term describes the dynamic power dissipation, the power dissipated at each transition of the system, where f is the system frequency, C is the average load capacitance per gate, a is the number of gates switching per clock period and V_{DD} is the power supply voltage. The second term is the direct current consumption which is due to the short circuit current I_{sc} and is less then 10% of dynamic power for high supply voltages [Pig98]. The last term is the static power dissipation which is caused by the average leakage current $I_{leak} = I_s \cdot e^{-V_T/n \cdot U_T}$.

Power consumption in analog circuits

Power dissipation in an analog circuits is generally lower than in digital ones, for low precision levels. Analog circuits dissipate much less switching energy, and the static power dissipation is proportional to Vdd.

A comparison established in [Kem97] between similar image sensors, one digital (CCD[1] camera with CDS[2]) and the other analog (APS[3]) shows the difference in power consumption. For the same pixel size (74 μm) and resolution (640x480 pixels) the CCD array has a 60 mW consumption and the $CMOS$[4] array 110 mW. However, when the peripheral circuits required

[1] CCD: Charge-Coupled Device. See glossary.
[2] CDS: Correlated Double Sampling. See glossary.
[3] APS: Active Pixel Sensor. See glossary.
[4] CMOS: Complementary Metal-Oxide-Semiconductor. See glossary.

to access the data (timer, driver, automatic gain control and ADC for the CCD chip and timer and ADC for the CMOS) are accounted, the analog chip dissipates 270 mW against 800 mW by the CCD.

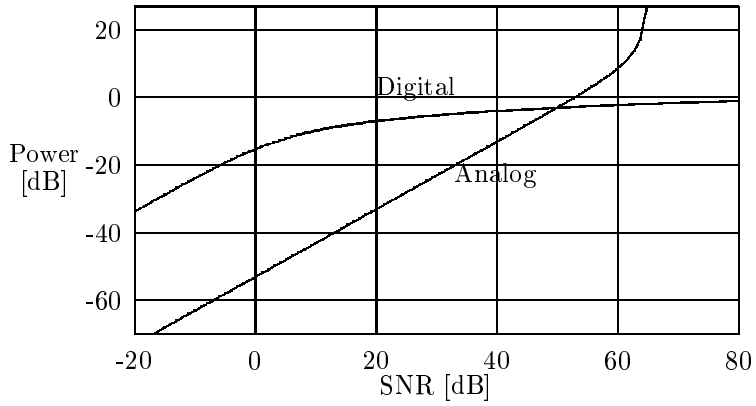


Figure 2.1: Power vs. SNR in digital and analog systems. From [SAR98].

Figure 2.1 shows the power dissipation as a function of the system precision (Signal-to-Noise Ratio) for digital and analog systems. As it can be seen, for low precision requirements, as it is usually the case in perceptive tasks, analog systems dissipate less power than digital ones. However, for high precision tasks such as restitution, digital is more convenient [ARR97]. The crossover point, where analog and digital systems dissipate the same power for a given precision depends on the technology and the designer's ability. For common, modern circuits, crossover is at around 8 bits, that is, a 48 dB SNR [SAR98].

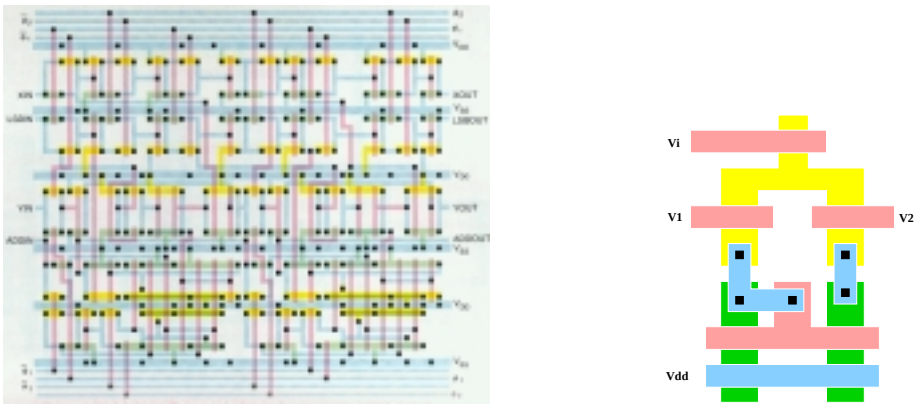
2.2.3 Size

When a low precision is sufficient, analog circuits also tend to be smaller in size than their digital equivalents, although it depends on the actual computation being carried. An analog multiplication using a *Gilbert multiplier* (cf. page 18) requires 8 transistors whereas a digital implementation, an 8-bit multiply, needs 3000 transistors [SAR98]. The comparison holds not only for computation, but also for communication; in the analog domain, one wire can transmit all the information, whereas in the digital world we need to have as many wires as bits.

The advantage of analog to digital systems varies with the function being implemented, since some are easily mapped to VLSI (e.g., a sum can be created by connecting two wires) and other functions are much more difficult to map (sine, for instance.) But in general terms, analog systems require less area for computation and communication [ARR94].

2.2.4 Processing and integration

The limited size of analog systems and the exploitation of the physical relations of their elements afford the integration of preprocessing in the first stages of perception. On the contrary, digital



(a) (b)

Figure 2.2: Layouts of (a) a digital and (b) an analog multiplier. (Not at the same scale).

systems tend to separate the actual sensor, the focal array in the case of vision sensors, from the A/D converter and the processing subsystem, typically a DSP or an application-specific circuit. This separation is sometimes imposed by technology: most CCD processes are incompatible with standard CMOS processes, and therefore no processing circuits can be integrated in the same wafer as the sensor [Kem97]. All this leads to digital systems being multichip ones, which are necessarily bulkier and more prone to interconnection problems than single-chip systems.

Analog sensors that are single-chip systems have a further advantage for mobile robotics: since part of the signal processing can take place on the sensor itself, they can relieve the robot's main processor from some burdensome computation, freeing it for more complicated tasks, or allowing it to switch into an idle mode.

2.2.5 Costs

It is very difficult to compare costs between digital and analog VLSI sensors, since most of the latter have been developed in academia, where cost was not an issue. However, several arguments can be brought forward, as well as an example.

Development costs

Analog VLSI circuits are developed by exploiting the features, the physical properties of the basic VLSI component, the transistor. For two different problems, different transistor properties may be used, thus leading to different designs. If we need a switch, we will use a transistor in saturation mode (cf. section 2.4.1), but if we want to generate a $log()$ function we will use it in weak inversion and it would then have a different size and other physical properties. Such an opportunistic design method precludes the development of advanced, automated electronic design (*EDA*) tools capable of dealing with analog circuits, be it from the point of view of layout, modeling, testing or hardware description languages. Digital design, on the contrary, benefits from a high level of design automation (through *VHDL*) and reusability (through the

standard-cell libraries), and therefore development costs are much lower.

Opportunistic design is not the only reason for the higher development costs of analog VLSI systems. As most of the VLSI circuits developed since the 70s are digital, it is normal that digital electronics design tools should be ahead of their analog counterparts.

Production costs

Production costs for analog vision sensors are lower than those for CCD-based digital cameras. The CMOS microprocessor fabrication processes are quite similar to the image-sensor ones and therefore benefit from a mainstream process which is supported by a large number of fabs (around 1000 [Die97]), as opposed to the around 15 fabs offering the CCD process.

However, color filtering and microlens deposition require additional steps that are not present in the standard CMOS process, and most foundries need to have an *implant optimization* to improve sensor quality. Nonetheless, for most CMOS circuits the advantages outweigh the limitations.

Industrial examples.

A successful example of a CMOS sensor-based industrial application does exist. Logitech SA (Romanel, Switzerland) has currently a whole line of trackball pointing devices based on the Marble system [Arr96]. These devices use the "inverted mouse" paradigm, whereby a red ball with dark spots is moved by the user's fingers, and a CMOS sensor and processor senses the 2D direction of motion and outputs a value proportional to the displacement. The sensor was developed at CSEM SA (Neuchâtel, Switzerland) and it has certainly benefitted from the economies of scale of mice production.

Figure 2.3: The Logitech Marble Mouse, a successful industrial application of A-VLSI technology.

Another example of industrial if not commercial CMOS camera is the VTS camera [IME97], implemented in CMOS APS technology by IMEC (Leuven, Belgium). It was launched in October 30th, 1997 on board the Ariane 502 mission, and it was responsible for tracking the payload separation.

2.2.6 Neuromorphism

A secondary advantage of analog VLSI, but which is of utmost importance in some academic research circles, is A-VLSI's ability to mimic some features of biological neurons, such as massive parallelism, real-time processing and collective computation. This affords a very interesting exchange between engineering and biology, whereby biologists can verify their models of neural computation on silicon, and engineers can apply the extremely efficient algorithms of neural systems to electronics. For instance, the neurophysiological model of the retina [Bea94] [Car81] has been used to embody vision sensors with the light-adaptation properties of the latter [Mah92] [Mea89] [And95]. Similarly, researchers are applying the known facts about the working of the brain [Hub94] [Chu92] to VLSI processing [Ind97].

2.3 Disadvantages of analog systems

Notwithstanding all the previously discussed advantages, analog sensors do have some difficulties that have to be weighted when choosing between an analog (CMOS) or digital (CCD) sensor.

2.3.1 Lack of precision

The main drawback, the one that most often comes out, is analog circuits' lack of precision. As it can be seen in figure 2.1, analog systems have an asymptotic limit due to $1/f$ noise, whereas digital systems can keep increasing the number of bits, therefore improving the precision (SNR), albeit at a great power and area cost.

It can be argued, however, that perceptive tasks do not need a great precision [Arr94], and therefore analog circuits are suitable for them.

2.3.2 Lack of flexibility

Analog VLSI circuits are etched and deposited on the silicon wafers used in CMOS fabrication. The circuit is "hardwired" and it cannot be modified nor reconfigured. Thus, an A-VLSI motion detector can not be used for edge detection. Digital systems based on microprocessors and DSPs can be modified by reprogramming the processing unit. At most, analog systems can tuned, that is their parameters can modified. These parameters are usually defined by reference currents or voltages, and they can control such aspects as cutoff frequencies in filters, velocity ranges in motion detectors or thresholds of activation in edge detectors.

The lack of flexibility in A-VLSI is also present in the access to signals. Because of the packaging technique, analog VLSI sensors are used as "boxes", with clearly defined outputs. Internal signals, (e.g., the photodetector outputs) can not be accessed unless an access was foreseen for that purpose. Thus inspection of "intermediate variables", as in computer programming, is impossible.

2.3.3 Interface

Almost all current mobile robots are digital, if only to control the motors and actuators. Interfacing analog sensors with these digital systems can be somewhat difficult.

Difficulties arise from three points. First, an analog subcircuit has to be dealt with, and that imposes some constrains on the circuit board design, since separate ground lines and a clean power supply are needed. Second, at a certain point an A/D converter is needed, but it is rarely very complicated, as it rarely exceeds 8 bits. In fact, as mentioned in section 2.2.4, A/D converters can be integrated in the sensor itself, giving it a digital output. Third, analog circuits require bias or reference currents (more often than voltages) that complicate the interface, as they are often in the microampere range. It has to be kept in mind however, that CCDs require several power supplies, which complicate interfacing too.

2.4 A-VLSI building blocks

Digital VLSI systems have been in use for nearly three decades now, and this has lead to the development of standard techniques and software tools for circuit design. Digital systems use transistors in *strong-inversion*, which can be viewed as boolean switches. Thus boolean logic is used to describe the function to be synthesized as a combination of simpler subfunctions that can be boolean operators such as AND or NOT. Each of these operators has an optimized structure that can be reused for the next design. These basic elements which can be used and connected together to make up the desired function are called *standard cells*.

None of this exists in analog VLSI circuit design. We take advantage of different transistor properties and therefore we need to design the transistors accordingly. Since these properties depend on physical and technological variables, the designs can be rarely reused. However, some basic structures can be found time and again in literature that produce some interesting functions. These *analog VLSI building blocks* shall be reviewed in the following sections.

2.4.1 The elementary brick: the transistor

The MOS[5] transistor is the most elementary device in VLSI circuits, and the most important. It can be an N-channel or P-channel transistor, depending on the type of carriers flowing through the channel.

In N-channel transistors such as the one shown in figure 2.4, electrons flow from source (S) to drain (D) according to the potentials present at the source, drain and gate (G). The drain current I_D is expressed as the difference of two currents (cf. equations 2.2 and 2.3).

$$I_D = I_F - I_R = I_S \cdot (f(V_G, V_S) - f(V_G, V_D)) = \tag{2.2}$$

$$I_D = I_S \cdot \left(\ln^2(1 + exp(\frac{V_G - V_{T0} - nV_S}{2nU_T})) - \ln^2(1 + exp(\frac{V_G - V_{T0} - nV_D}{2nU_T})) \right) \tag{2.3}$$

where V_{T0} is the *gate threshold voltage* with a typical value of 0.5 to 1.2 volts. U_T is a constant equal to $\frac{kT}{q} = 26mV$ at 300^o K. Parameter n is the *slope factor*, with a typical value of 1.2 to 1.6 [Vit96] and I_S is the *specific current* of the transistor, which depends on technological factors according to

$$I_S = 2n\mu C_{ox} \cdot U_T^2 \cdot \frac{W}{L} \tag{2.4}$$

[5] Metal-Oxide-Semiconductor

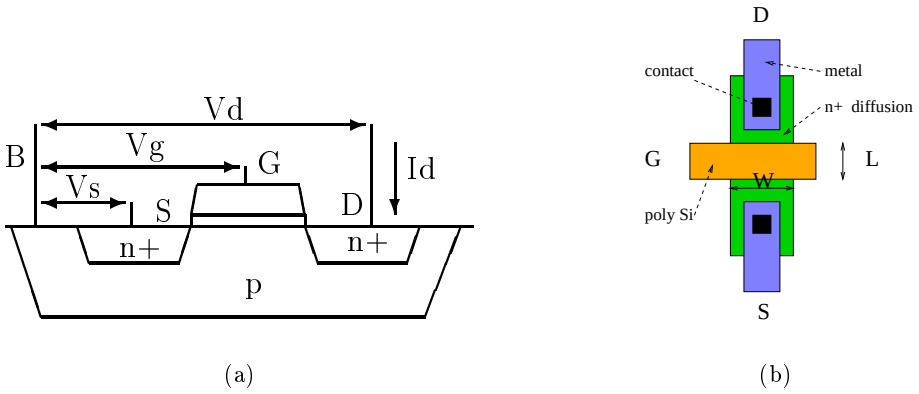


Figure 2.4: (a) Cross-section of an N-channel MOS transistor and (b) its layout.

where μ is the carrier mobility in the channel, C_{ox} is the gate oxide capacitance per unit area and W and L are the width and length of the channel. Typical specific currents for minimum-size transistors are in the 20 nA to 200 nA range.

N-channel transistors are represented in circuit schematics by the symbol in figure 2.5a. A symmetrical device, the P-channel transistor (cf. figure 2.5b) exists, and it abides by the same equation 2.3.

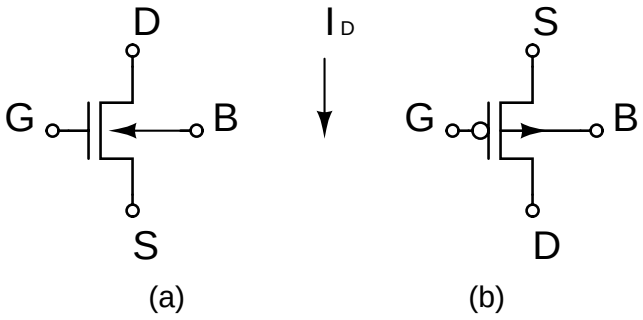


Figure 2.5: Schematic symbols of (a) N-channel MOS transistor and (b) P-channel MOS transistor.

The MOS transistors can be in four modes of operation:

1. In *conduction*: When both terms in equation 2.3 are in *strong inversion* mode, also called *above threshold* mode. This happens when V_S and V_D are below the *pinch-off voltage* $V_P = \frac{V_G - V_{TO}}{n}$. Under these circumstances, equation 2.3 can be rewritten as

$$I_D = \frac{I_S}{2nU_T^2}(V_D - V_S)\left(V_G - V_{T0} - \frac{n}{2}(V_D + V_S)\right) \tag{2.5}$$

2. In *saturation*: when one of the terms in equation 2.3 is in strong inversion and the other is negligible, the drain current becomes

$$I_D = \frac{I_S}{(2nU_T)^2}\,(V_G - V_{T0} - nV_S)^2 \tag{2.6}$$

if $V_S < V_P < V_D$ or else

$$I_D = -\frac{I_S}{(2nU_T)^2}\,(V_G - V_{T0} - nV_D)^2 \tag{2.7}$$

if $V_D < V_P < V_S$.

3. In *weak inversion*: When both terms in equation 2.3 are in *weak inversion* mode [VIT94], also called *subthreshold* mode [MEA89]. This happens when both V_S and V_D are greater than V_P. In this situation, equation 2.3 becomes

$$I_D = I_S \cdot e^{\frac{V_G - V_{TO}}{nU_T}} \cdot \left(e^{\frac{-V_S}{U_T}} - e^{\frac{-V_D}{U_T}}\right) \tag{2.8}$$

   This mode is particularly interesting because of the exponential relationship between I_D and V_G, V_S, V_D. *Weak inversion* is therefore the most common mode of operation in analog, neuromorphic VLSI circuits.

4. *Blocked*: When both current components are smaller than leakage currents, the transistor is said to be blocked.

2.4.2 Simple operators

Signals in VLSI circuits are represented by physical magnitudes such as voltage, current, charge or frequency. Each representation has its advantages and drawbacks. Voltage is very easy to distribute or copy, since it suffices to pass a wire. On the other hand, it consumes a large amount of energy because it need to charge and discharge the node parasitic capacitance and it is very difficult to add or subtract.

Current is of course easier to add; Kirchhoff's law states that just connecting two wires is enough to do so. However, current distribution is complicated, since it requires copying.

Charge is also a signal easy to process with switched-capacitors. Its main drawback is that these circuits introduce time-sampling, which is undesirable in most cases.

A fourth possible representation is pulse-frequency or pulse-width modulation. This technique is very convenient for communication systems and it has the additional interest of being the most common in biological neural systems, but it also introduces time sampling and a complicated circuitry [VIT94].

Most of the building blocks discussed in the following subsections are *current mode* circuits. Nonetheless, voltage-mode equivalents of most of them do exist.

Weighted copy

The weighted-copy operation is needed for current-mode signals, since their distribution requires the replication of a given current. The structure that performs this operation, called *current mirror* is shown in figure 2.6.

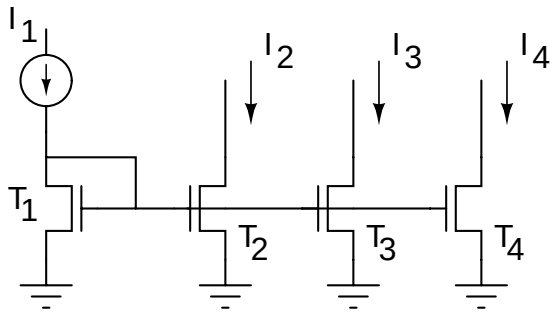


Figure 2.6: NMOS current mirror: Currents I_2, I_3 and I_4 are a copy of the positive part of I_1.

If transistor T_2 is in saturation ($V_{D2} >> V_{Dsat}$ and I_2 is independent of V_D) the two transistors T_1 and T_2, having common source and gate voltages have a current ratio of

$$\frac{I_2}{I_1} = \frac{I_{S2} \cdot e^{\frac{V_G - V_{T0}}{nU_T}} \cdot e^{\frac{-V_S}{U_T}}}{I_{S1} \cdot e^{\frac{V_G - V_{T0}}{nU_T}} \cdot e^{\frac{-V_S}{U_T}}} = \frac{I_{S2}}{I_{S1}} = \frac{(W/L)_2}{(W/L)_1} \tag{2.9}$$

Thus, by setting up the widths and lengths of the transistors accordingly we can define the weight of the copy, which can only be a rational number. This is however prone to errors, particularly in weak inversion, because of mismatches in threshold voltages (V_{T0}) and, to a lesser degree, device dimensions. It is recommended to use identical transistors, connecting them in parallel to reach the desired ratio [Vit94].

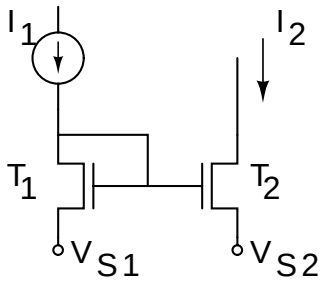


Figure 2.7: NMOS current mirror with variable weight (source bias).

Current mirrors such as those in figure 2.6 imply a lack of flexibility, since the current ratio cannot be changed at "run-time" as in a fine tuning. An alternative architecture exists, the current mirror with source bias, whereby the transistors have different source voltages, as shown in figure 2.7. In this case, the weight is controlled by the source voltages according to

$$\frac{I_2}{I_1} = e^{\frac{V_{S2}-V_{S1}}{U_T}} \tag{2.10}$$

provided that $V_{S1} > V_{S2}$.

Addition

Addition in current mode is as simple as connecting the input wires to the output wire at a common node. However, in some cases it is desirable to do so without changing the voltage at the common input node. A circuit called *current conveyor* (figure 2.8) combines some current mirrors to replicate the input current while keeping the input node at a fixed potential [VIT94].

A simplified two-transistor current conveyor is proposed in [AND91].

Multiplication

Multiplication, that is, variable amplification, can be achieved by several circuits. In the next pages two such solutions will be described.

Transconductance amplifier The first one, called *transconductance amplifier*, is based on a *differential pair*: two transistors with a common source are biased with a constant, total current (cf. lower half of figure 2.9a). If both transistors are in weak inversion and saturated, the current flowing through each one is exponentially dependent on the gate voltage and we obtain the following equation:

$$I_1 - I_2 = I_b \cdot \tanh(\frac{V_1 - V_2}{2nU_T}) \tag{2.11}$$

If the differential voltage is sufficiently small compared to $2nU_T$, then $\tanh(x) \approx x$ and equation 2.11 can be approximated as

$$I_1 - I_2 = I_b \cdot \frac{V_1 - V_2}{2nU_T} \tag{2.12}$$

which yields a multiplication of current I_b by the differential voltage $\Delta V = (V_1 - V_2)$. To obtain the desired current difference we use a current mirror (cf. top of figure 2.9a) so that $I_o = I_3 - I_2 = I_1 - I_2$.

Transconductance amplifiers are indeed very practical structures, but they present some major drawbacks. The linear region of operation can be very small, and the output voltage range is not symmetrical. The maximum output voltage can reach the supply voltage V_{dd}, but the minimum output cannot reach $-V_{dd}$ due to device mismatch.

Even in the case of an ideal transconductance amplifier, the multiplication can not be signed, since I_b can only have positive values. If a fully-signed multiplication (also called four-quadrant multiplication) is required, a circuit called *Gilbert multiplier* [MEA89] [GRA93] should be used. The Gilbert multiplier separates each signal in positive and negative components, producing two

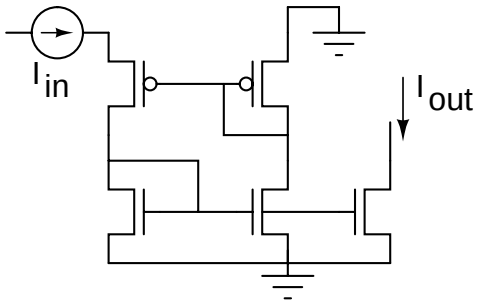


Figure 2.8: Current conveyor

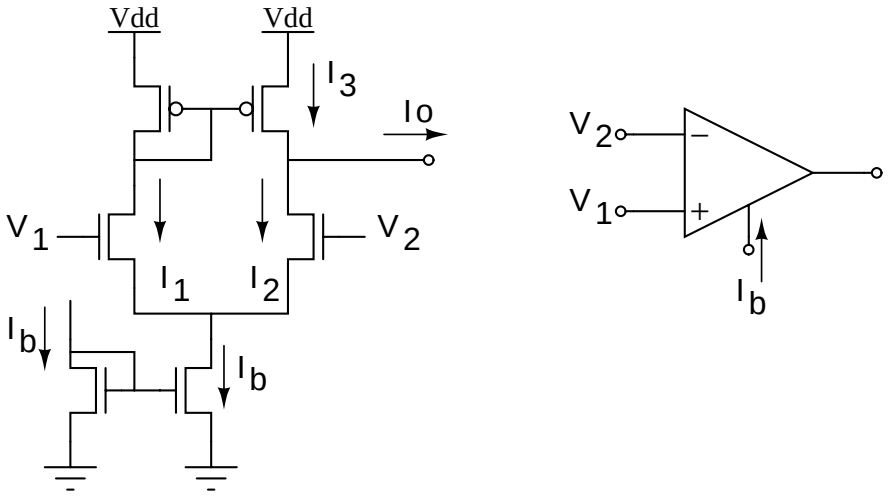


Figure 2.9: Transconductance amplifier: (a) structure and (b) schematic symbol.

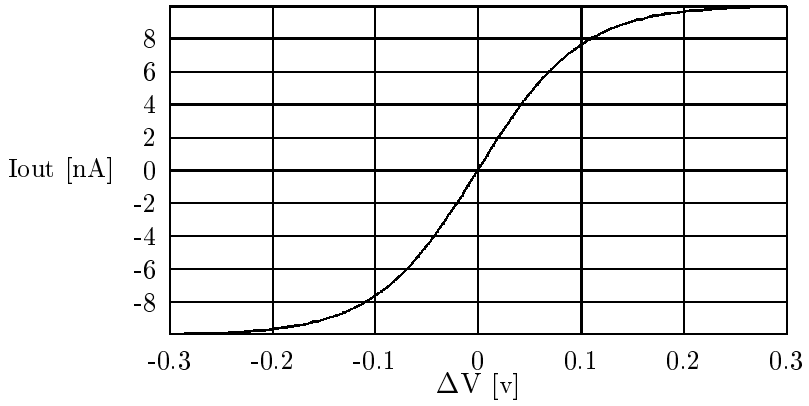


Figure 2.10: Transconductance amplifier response. The central, linear region ($\Delta V << 0.1$) can be used for multiplication/amplification, whereas the saturation region ($\Delta V >> 0.1$) can be used for output clamping or to obtain the *signum* function.

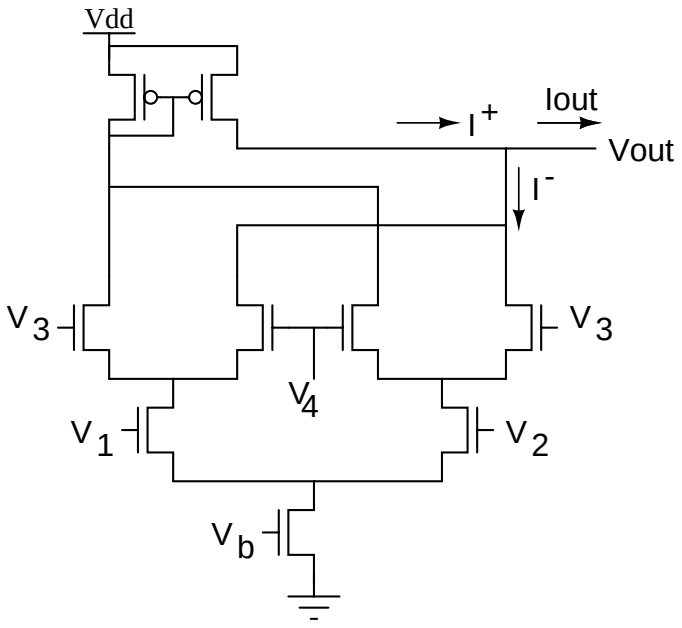


Figure 2.11: The four-quadrant or Gilbert multiplier can be used to obtain signed multiplication.

separate multiplications whose results are subtracted to obtain the total result.

The behavior of the Gilbert multiplier in figure 2.11 is described by

$$I_{out} = I_b \cdot \tanh(\frac{V_1 - V_2}{2nU_T}) \cdot \tanh(\frac{V_3 - V_4}{2nU_T}) \overset{\left\{ \begin{smallmatrix} V_1 - V_2 \ll 1 \\ V_3 - V_4 \ll 1 \end{smallmatrix} \right\}}{\Longrightarrow} I_{out} = I_b \cdot \frac{V_1 - V_2}{2nU_T} \cdot \frac{V_3 - V_4}{2nU_T} \quad (2.13)$$

Translinear loop The second structure that can be used to obtain a multiplication is called *translinear loop*; this is a closed loop of CMOS transistors in weak inversion mode, working as translinear devices, that is devices having a linear relationship between transconductance and current [AND91]. Figure 2.12 shows such a structure, with the following Kirchhoff's voltage law:

$$V_1 + V_2 + V_3 + V_4 = 0 \quad (2.14)$$

By replacing the subthreshold drain current (cf. eq. 2.8) into each of the gate-source voltages in equation 2.14 we have

$$V_{T0} + nU_T \cdot \ln(\frac{I_x}{I_s}) + V_{T0} + nU_T \cdot \ln(\frac{I_y}{I_s}) - V_{T0} - nU_T \cdot \ln(\frac{I_w}{I_s}) - V_{T0} - nU_T \cdot \ln(\frac{I_z}{I_s}) = 0 \quad (2.15)$$

which simplifies to

$$\ln(\frac{I_x}{I_s}) + \ln(\frac{I_y}{I_s}) - \ln(\frac{I_w}{I_s}) - \ln(\frac{I_z}{I_s}) = 0 \quad (2.16)$$

Therefore, the translinear loop has the following equation:

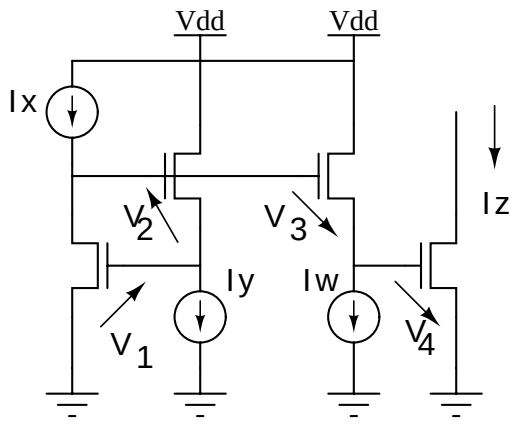


Figure 2.12: A translinear loop performs a multiplication/division operation.

$$\ln(\frac{I_x \cdot I_y}{I_s}) = \ln(\frac{I_w \cdot I_z}{I_s}) \Rightarrow I_z = \frac{I_x I_y}{I_w} \tag{2.17}$$

This makes the translinear loop a fully current-mode operator, as both inputs and outputs are currents.

Division

Current-mode division can be accomplished in two ways. The direct one is by using the *translinear loop* discussed above. The indirect one uses a feed-back loop to turn the division into a multiplication. If we want to obtain the following division

$$\frac{A}{B} = C \tag{2.18}$$

we can turn the equation into

$$A = B \cdot C \Rightarrow A - B \cdot C = error \tag{2.19}$$

The feedback loop makes the *error* signal tend to zero. Such a stratagem has been successfully used in Tanner and Mead's motion detector [TAN89], and Moore and Koch's variant of the latter [MOO91].

2.4.3 Nonlinear operators

Based on equation 2.3 it follows that a single transistor can generate nonlinear functions such as square, square-root, exponential, and logarithm functions [VIT94]. Indeed, in *strong inversion* the transfer function $I(V_g)$ is quadratic (cf. equation 2.6) and a square root voltage can be obtained by imposing a current I through the transistor. Conversely, a square current can be obtained by imposing the gate voltage. In weak inversion the transfer function is of exponential nature, and therefore an exponential current or a logarithmic voltage can be obtained.

2.4.4 Resistive networks

Resistive networks are current-mode, linear circuits made up of arrays of resistors and current sources whose input signals are "diffused" throughout the circuit. Resistors can be created in silicon by using metal or better, silicon oxide layers. This is however not very practical, because the resistance cannot be electrically controlled. Moreover, very long resistive paths would be needed to obtain the desired values, because of the low square-resistance of the silicon masks in current processes.

Pseudoconductance

CMOS transistors have a nice relationship between the gate voltage and the current flowing through them. If we define an artificial voltage, a *pseudovoltage*

$$V^* = -V_o \cdot f(V, V_G) \tag{2.20}$$

with a scaling voltage V_o then we can write equation 2.2 as

$$I_D = I_S \cdot (f(V_G, V_S) - f(V_G, V_D)) = I_S \cdot \left(-\frac{V_S^*}{V_o} + \frac{V_D^*}{V_o}\right) = \frac{I_S}{V_o}\left(V_D^* - V_S^*\right) \tag{2.21}$$

which becomes

$$I_D = g^*\left(V_D^* - V_S^*\right) \tag{2.22}$$

Equation 2.22 is Ohm's law, a linear relationship between the current through a resistor and voltage across it. The ratio I/V defines the resistor's conductance. Since we have an artificially created resistor we shall refer to the transistor's *pseudoconductance* $g^* = \frac{I_S}{V_o}$ [Vit94].

Some interesting properties follow from equation 2.22:

1. In a resistor network, we can replace the resistors with transistors interconnected by their sources and drains with a common gate voltage and obtain a linear circuit with respect to the currents [Vit94].

2. Since function $f(V, V_G)$ in equation 2.3 is monotonically decreasing and nonnegative we can conclude that *pseudovoltage* V^* cannot change sign, and *pseudoground* 0^* will be reached for large (saturating) values of the real voltage V.

3. The existence of such *pseudoground* affords the measurement of the current flowing through the pseudoresistor by means of a current mirror [Vit94] [Mea89].

4. Normally the pseudoconductance depends on physical parameters of the transistor, since $g^* \propto I_S \propto W/L$.

5. All transistors must have a common function $f(V, V_G)$ and therefore they must all have a common gate voltage V_G.

In weak inversion, or subthreshold region, $f(V, V_G) = e^{\frac{V_G - V_{T0}}{nU_T}} \cdot e^{-\frac{V}{U_T}}$ and therefore the pseudovoltage and pseudoconductance can be redefined as

$$V^* = -V_o \cdot e^{-\frac{V}{U_T}} \tag{2.23}$$

$$g^* = \frac{I_S}{V_o} \cdot e^{\frac{V_G - V_{T0}}{nU_T}} \tag{2.24}$$

Equation 2.22 remains valid (and linear in currents) but now the pseudoconductance g^* is variable and controlled by V_G. Since all transistors have a common reference 0^* that does not depend on g^*, gate voltages can be changed independently: the variable pseudoresistor is born. The resistance of this pseudodevice is defined by

$$R^* = \frac{V_o}{2nU_T^2 \mu C_{ox}} \cdot e^{\frac{V_{T0}}{nU_T}} \cdot e^{-\frac{V_G}{nU_T}} \tag{2.25}$$

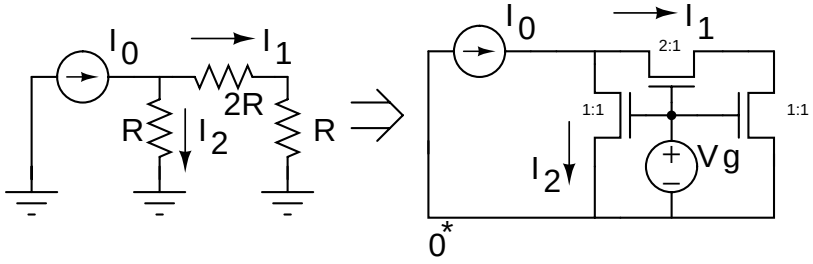


Figure 2.13: A linear resistive network can be replaced by a transistor network. For currents only, transistors in weak-inversion mode behave as resistors with pseudoresistance R^* defined by equation 2.25.

In the following subsections, resistive networks in general will be discussed. For simplicity, supportive figures will show resistors, but they correspond to actual transistors in the implementations.

Resistive diffusion networks: analysis

The most common application of VLSI pseudoresistors are resistive diffusion networks. The simplest example of diffusion network is shown in figure 2.14.

In a given network section like in figure 2.15 we have the following equations:

$$\begin{aligned} i[n] \cdot R &= v[n] - v[n+1] && (2.26) \\ G \cdot v[n+1] &= i[n] - i[n+1] + i_{in}[n+1] && (2.27) \end{aligned}$$

By applying the Z-transform we obtain

$$\begin{aligned} IR &= V - V \cdot z && (2.28) \\ GV \cdot z &= I - I \cdot z + I_{in} \cdot z && (2.29) \end{aligned}$$

By replacing eq. 2.28 into eq. 2.29 we obtain the characteristic equation

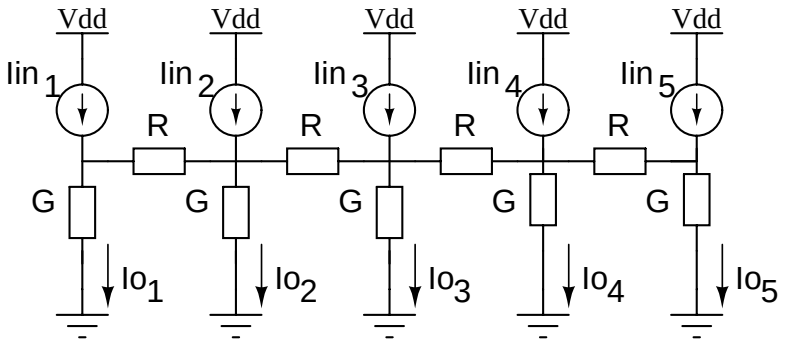


Figure 2.14: A generic resistive network: The output currents I_o are measured on the vertical (G) resistors. They are due to the diffusion of the current I_{in} injected at each node through the horizontal (R) and vertical resistors.

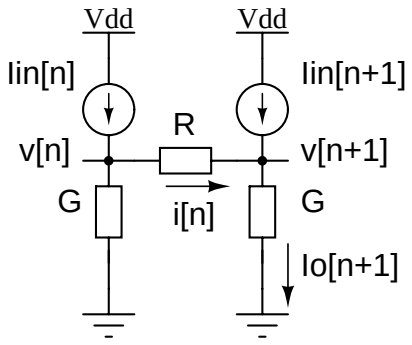


Figure 2.15: A section of a generic resistive network.

$$V\left[z^2 + (2 + RG)z + 1\right] = 0 \tag{2.30}$$

whose solution is

$$z = 1 + \frac{1}{2\lambda^2} - \frac{1}{\lambda}\sqrt{1 + \frac{1}{4\lambda^2}} \tag{2.31}$$

where $\lambda = \frac{1}{\sqrt{RG}}$.

If we place a current source i_{in} at the origin, the output current $i_o[n]$ measured n nodes away will have the form

$$i_o[n] = G \cdot v[n] = G \cdot (v[0]z^n) = G \cdot (i_{in}[0]R_{in}z^n) \tag{2.32}$$

$$\frac{i_o[n]}{i_{in}[0]} \propto z^n = e^{n \ln z} \tag{2.33}$$

Applying the Taylor approximation $\ln(1+x) \approx x$ for small x to the logarithm of eq. 2.31 we have

$$\ln(z) \approx \frac{1}{2\lambda^2} - \frac{1}{\lambda}\sqrt{1 + \frac{1}{4\lambda^2}} \approx -\frac{1}{\lambda} \tag{2.34}$$

for large values of λ. Hence the transfer function of a resistive network approximates

$$\frac{i_o[n]}{i_{in}[0]} \propto z^{|n|} = e^{-\frac{|n|}{\lambda}} \tag{2.35}$$

where λ is the *diffusion length*, which can be electrically controlled by the gate voltages of the vertical and horizontal pseudoresistors, according to eq. 2.25.

Resistive diffusion networks: filters

Since diffusion networks are linear circuits, we can apply the *superposition principle* to equation 2.35 to obtain the response of the diffusion network in figure 2.14

$$I_o(n) = GR_{in} \sum_k e^{-\frac{|k-n|}{\lambda}} I_{in}(k) \tag{2.36}$$

Equation 2.36 has a spatial low-pass filtering profile with cut-off frequency $\frac{1}{\lambda}$, which can be electronically varied. It produces a local average, a smoothed version of the input signal. As such, it is often used to approximate the *gaussian filter*, the most common smoothing filter used in computer vision [Row96] [Mar82] [Mea89] [Moi96].

Diffusion networks can be combined to obtain other filter responses. Thus for instance the subtraction of two smoothed-out responses yields a band-pass filter whose impulse response is the well-known *Mexican hat function* [And95]. Such a response has been detected in the outer plexiform layer (OPL) of the retina [Bea94] [Kra99] and it is referred in neurophysiological literature as *center on-surround off* response (figure 2.16).

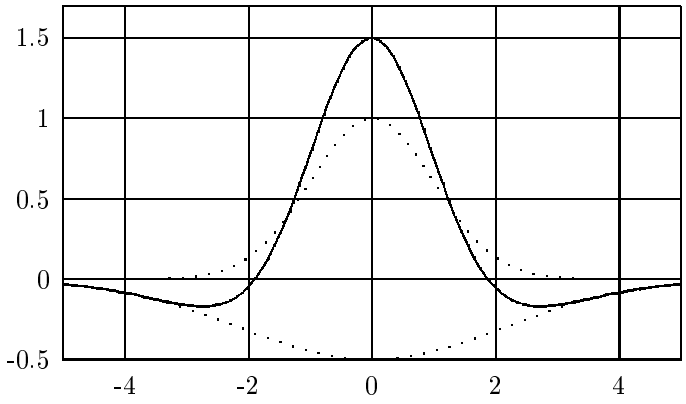


Figure 2.16: The *center on-surround off* response, also known as Mexican hat function, can be obtained by subtracting the output of two resistive networks.

A highpass version of the input signal can be obtained by subtracting the lowpass output from the original image, as in Mead's "Silicon Retina" [MEA89].

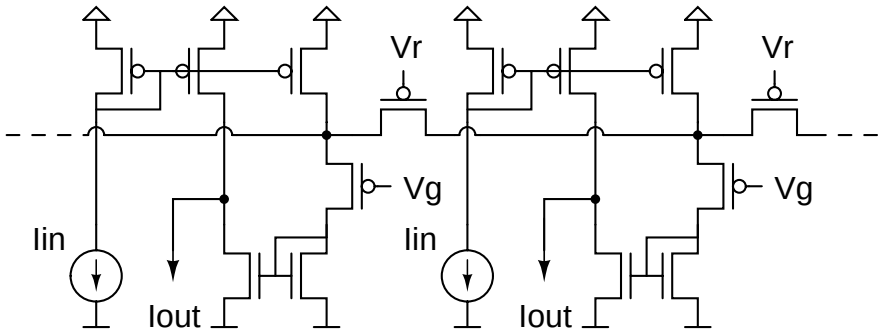


Figure 2.17: Pseudoconductance implementation of a highpass filter. From [VIT96].

A particularly interesting resistive network appears when $G = 0$ and we measure the current flowing at the extremities by means of current mirrors [VEN96]. Suppose we have a circuit made of N horizontal resistors R, as in figure 2.18. At each node a current J_i is injected, and the current source 'sees' an input resistance R_{1i} to the left and R_{2i} to the right. The current at the right extremity will be

$$I_{o2} = \sum_i \frac{V_i}{R_{2i}} = \sum_i \frac{V_i}{R(N-x_i)} = \sum_i \frac{1}{R(N-x_i)} J_i \frac{R_{1i}+R_{2i}}{R_{1i}R_{2i}} = \tag{2.37}$$

$$= \sum_i \frac{1}{R(N-x_i)} J_i \frac{Rx_i R(N-x_i)}{Rx_i + R(N-x_i)} = \sum_i J_i \frac{x_i}{N} \tag{2.38}$$

$$\tag{2.39}$$

and of course, Kirchhoff's law states that

$$I_{o1} + I_{o2} = \sum_i J_i \tag{2.40}$$

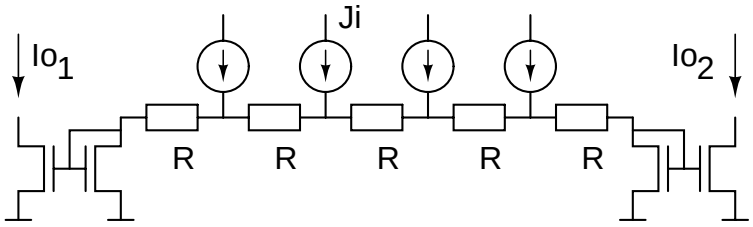


Figure 2.18: Resistive network that calculates the center of gravity of the input distribution $J(i)$. From [VEN96].

Therefore the output current ratio

$$\frac{I_{o2}}{I_{o1}+I_{o2}} = \frac{\frac{1}{N}\sum_i J_i x_i}{\sum_i J_i} \tag{2.41}$$

yields the center of gravity of the signal distribution $J(i)$.

2.4.5 Collective computation

If there is a class of operations at which analog VLSI clearly excels above digital or even digital VLSI implementations, it certainly is *collective computation*. Collective can be described as the calculation of a single value, usually the output, based on several inputs by means of feedback loops between the individual sources and output value. The feedback loops allow for a constant "correction" of the input values, until a satisfying solution is found.

In general, collective computation circuits consist of a number of cells that share a certain number of signals. A voltage is therefore the common signal, since it is simple to propagate to each cell. The common voltage V_c is a function of the output currents x_i of each cell. In turn, the cell's output current is a function of the common voltage and the cell's input voltage or current y_i. Mathematically we have

$$V_c = f(\vec{x}) \tag{2.42}$$
$$\vec{x} = \vec{g}(V_c, \vec{y}) \tag{2.43}$$

which implies

$$V_c = f(\vec{g}(V_c, \vec{y})) = h(V_c, \vec{y}) \tag{2.44}$$

The circuit solves equation 2.44, which resembles a fixed-point iteration problem. VLSI offers, of course, the advantage of real-time constraint satisfaction. These principles of collective computation will be made clear by the examples that follow.

Weighted average: Follower aggregation

Weighted averaging is a very common operation when processing data. It can be used to filter out noisy data, to take a decision based on multiple experts, or to calculate the center of gravity of a signal. Such an operation can be performed in analog VLSI by means of the *follower aggregation* circuit, shown in figure 2.19.

Each cell in circuit 2.19 injects into the output node V_o a current proportional to the associated input voltage V_i. A feedback mechanism is present because the current is influenced by V_o. By applying Kirchhoff's law we have

$$I_o = 0 = \sum_i G_i \cdot (V_o - V_i) = V_o \sum_i G_i - \sum_i G_i V_i \tag{2.45}$$

which implies

$$V_o = \frac{\sum_i G_i V_i}{\sum_i G_i} \tag{2.46}$$

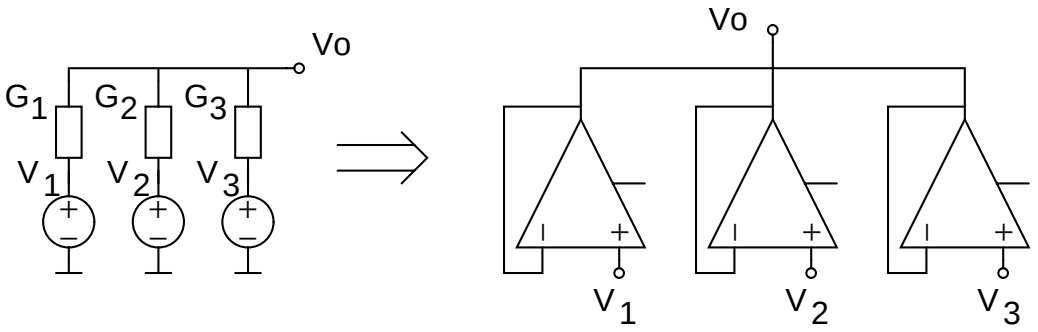


Figure 2.19: The follower aggregation circuit calculates the average of input signals V_i

The conductances G_i are usually implemented as transconductance amplifiers (cf. section 2.4.2), allowing for the weights G_i to be electrically changed by means of the amplifiers' bias currents I_b.

Winner-take-all

A very common operation in sensory data processing is picking the biggest value out of a set. In mobile robotics it may take place in a motion detection vision system: the fastest-moving target has to be picked out in order to react to the fastest-moving obstacle. This is not a $max()$ function, since we are not interested in the actual value of the input, but rather the position (the index) of the maximum value. Such an operation is called *Winner-take-all* because it is implemented as a competition among input data whereby the greater value chokes off the other inputs.

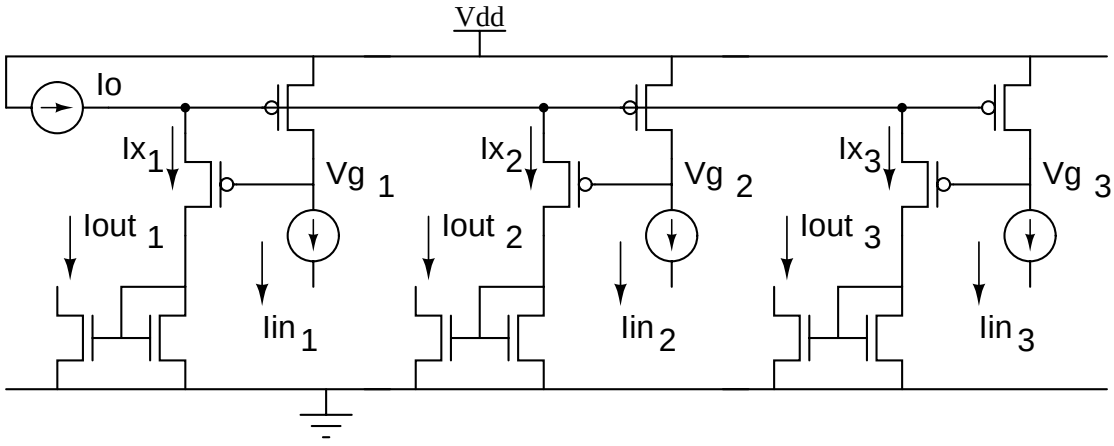


Figure 2.20: The *Winner-take-all* network identifies the cell with the highest input by setting $I_{\mathrm{out}\,i} = I_o$ for the cell $I_{\mathrm{in}i|max}$

The original implementation by Lazzaro et al. [Laz88] is shown in figure 2.20, but more evolved variants are discussed in [Ind97]. In the former circuit, the input currents $I_{\mathrm{in}i}$ drive the gate voltages V_{gi} down, draining more current I_{xi}. The cell with the highest input current

will drain more current, driving V_g lower, and draining in turn more current I_{xi}. This positive-feedback condition will stop when the winner's I_x reaches the limiting current I_o, chocking the other cells' I_x. At that point, the winner cell will have $I_{out}(winner) = I_o$ and all the other outputs will be $I_{out} = 0$.

2.5 A-VLSI Sensor Components

Most analog VLSI vision sensors have a common structure, irrespective of their functionality. In this section, the typical sensor components will be overviewed and discussed.

2.5.1 Optics and Sensing

For practical reasons, most A-VLSI sensors use standard optics. Therefore, standard CS or C mount optics are used. In some cases, such as the EDI retina [Lan96] special optics have been developed because of the panoramic view of the chip (cf. sec. 7.3).

By passing through the optics, input light is focused on the silicon surface, where photodetectors convert the photon flow into a flow of electrons, i.e., current. Any N/P junction is capable of transducing light into current, and therefore several types of photodetectors are available in standard CMOS processes, all of them actually based on parasitic devices [Moi96]. The simplest one is the *photodiode*, which comes in several "flavors" (n^+ to p substrate, p^+ to n-well, etc.) and has the disadvantage of having a low sensitivity. *Phototransistors* are parasitic vertical bipolar devices (pnp junctions); they have a higher sensitivity albeit a slower response.

New types of photodetectors have been developed in the last years by incorporating some photocurrent amplifying capabilities at each pixel site [CMO98]. These devices are called active photodetectors or APS (Active pixel sensors). Among them we can find active photodiodes, active phototransistors and *charge-modulated devices* (*CMD*), a hybrid between CMOS and CCD sensors [Kem97]. These active pixels require a larger area per pixel, and they are mostly used in pure imagers (without specific processing) that compete with CCD cameras.

2.5.2 Adaptation and normalization

Lightning conditions can change suddenly, more so on a mobile robot that can move from dark to light areas, a change of some 10 orders of magnitude. Some mechanism capable of adapting the sensor operating point is needed.

One adaptation mechanism consists in applying a logarithmic compression to the photocurrent. This is feasible in the subthreshold region, because of the exponential characteristic between voltage and current. A circuit such as in figure 2.21 produces a voltage that depends on the logarithm of the light intensity [Moi96].

This technique has the advantage of facilitating the extraction of invariant information in the image: contrast. Light intensity in an image is made of background illumination I_{bg} plus a small signal i that depends on the objects' reflectivity. Contrast is defined as the ratio of the signal to the background ($c = \frac{i}{I_{bg}}$), and it can be recovered from the variation in the output of

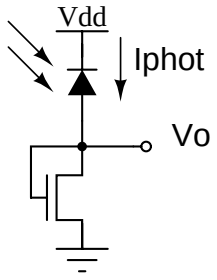


Figure 2.21: Photoreceptor with a voltage output proportional to the log of light intensity.

a logarithmic photodetector [DEL96]. (See eq. 2.47.)

$$\partial \log(I) = \frac{\partial I}{I} = \frac{i}{I_{bg}} \tag{2.47}$$

Still, a simple logarithmic photodetector has some problems such as slow response and device mismatch that preclude its use in less than ideal conditions. The best solution is the addition of a feedback loop capable of adapting the photodetector operating point. Delbruck's *adaptive photodetector* is described in detail in [DEL96] and it has been successfully used in several circuits [HAR98A] [KRA98].

Another solution to recover the contrast information without using an adaptive photodetector is to apply some processing further down the processing chain. This technique is called *algorithmic adaptation* by Moini [MOI96]. Since contrast is a ratio between the signal and the average intensity, one solution consists in dividing the photocurrent by the average photocurrent, as in [LAN96]. This technique fails when there is a bright spot on the image, since it biases the average. This can be improved by using the local average, by means of a resistive filter (cf. section 2.4.4) as in [VEN97].

2.5.3 Filtering and processing

The next element in an analog VLSI chip would be the processing itself, which can be purely spatial (e.g., edge detection [VEN97], object detection [VEN96], stereo matching [MAH89]), spatio-temporal (motion detection [KRA96], focus-of-expansion detection [SAR96], optic flow [STO98]) or purely temporal. An overview of analog VLSI vision chips can be found in [MOI96].

2.5.4 Output

After processing the image on the sensor itself, the information has to be passed on to another element in the system. This can be, either another analog-VLSI chip that performs some further processing (as in the brain's cortex) or else a digital microprocessor, such as the robot's CPU, that will use the sensor information to command the motors.

In the first case, it is desirable to communicate while remaining in the analog domain, because this avoids conversion overhead while conserving analog VLSI's advantages (cf. section 2.2). Wiring (area) costs in VLSI chips impose serial communication as the best solution.

This is facilitated by the sparse amount of data to be transmitted in some cases (eg. edges occupy a small percentage of an image area). Therefore, some asynchronous serial communication protocols have been proposed. A non-arbitrated version is proposed in [MOR94] and one using a bus arbiter in [MAH94].

In mobile robotics, however, the most common situation will be to access the vision sensor or the second-stage analog processing ("brain") with a digital processor, most probably with the robot's main CPU. Therefore some kind of analog-to-digital converter will be needed. In some cases an external ADC is used to convert an analog output. The main drawbacks of this approach are the increased noise in the measure because an analog signal has to be brought into a digital (noisy) environment, the difficulty in generating the adequate reference current or voltage (since the analog signal may run in the nanoamperes or microvolts) and the increased size that an additional chip supposes.

The most convenient solution for mobile robotics is the embedding of the ADC into the chip as in [MEY98] [LAN96]. As discussed in section 2.2.2, analog systems have lower power consumption and size than digital ones up to a certain level of precision (cf. figure 2.1) which is around 8 to 10 bits. In most cases therefore, an 8-bit ADC should be enough. It should be designed taking particular care not to consume much power and area (lest we loose the benefits of using A-VLSI) and avoiding deteriorating the image quality because of crosstalk or hot spots [FOS98].

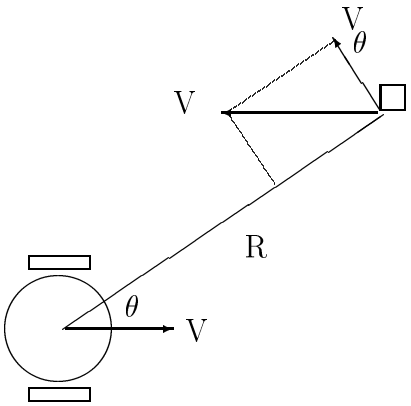
2.6 Examples

This chapter is completed by an overview of some applications of VLSI vision sensors to mobile robotics. The purpose is twofold: to show how the elements discussed in the previous sections (particularly sections 2.4 and 2.5) can be combined, and to see how mobile robots have benefitted from such sensors.

2.6.1 A fly-like motion-detecting retina

The fly's eyes consist of some 3000 pixels, each one containing eight photodetectors and its own lens. Because of this structure they are said to be *compound eyes*. The photodetectors are connected with their 7 neighbors in the pixel or facet which in turn is connected to the surrounding facets [FRA92]. These lateral interactions among photodetectors lead to a directionally-selective motion-detection mechanism that has been actually identified on the H1 neuron, located in the 3^{rd} optic ganglion. A sequential stimulation of the photoreceptors in the preferred direction produces an activation of this neuron. The fly's eye is therefore said to be made of *elementary motion detectors* (*EMD*). The flies use this motion information (the angular speed of the contrast point) together with its own motion to determine the distance to the contrast point. Objects with low angular velocity are considered to be farther away than objects with higher angular velocity. Such a technique is called *motion parallax*.

Franceschini et al. modeled these EMDs with *Reichardt cells*, that correlate a photoreceptor input with a delayed version of the neighboring photoreceptor. The EMDs were implemented with discrete analog circuits, and they had a nonuniform distribution, with a finer spacing towards the front, as in the fly. Such a distribution seems to compensate a sine law in the optic



$$V_\theta = R\frac{d\theta}{dt} = V \sin\theta$$

Figure 2.22: The principle of motion parallax: If velocity V is known, the distance R to the detected object can be calculated from the angle θ under which the object is seen.

flow equation [FRA92]. They placed an array of 100 such EMDs on a Real World Interface robot that moved around in small rectilinear steps at 50 cm/s. Such a motion allowed the robot to recover the translational component of the optic flow field and move freely in a cluttered environment.

In 1997 Harrison and Koch built a VLSI circuit based on the same model of the fly's motion detection. The structure of their EMD is shown in figure 2.23.

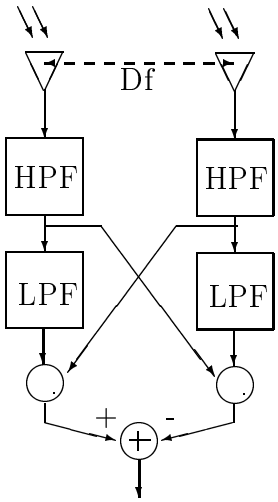


Figure 2.23: An Elementary Motion Detector inspired from the fly's eye. From [HAR98B].

The EMDs are composed of two processing chains starting with photoreceptors separated by an angle $\Delta\phi$. Next, a temporal highpass filter as the one described in page 26 is used to detect moving edges (contrast or luminance changes). The output of the highpass filter is multiplied with a delayed version of the neighboring signal. The delay is approximated by the inherent phase-lag of a temporal lowpass filter that can be tuned by means of an external bias volt-

age. The multiplier circuits shown in figure 2.23 are floating-gate FET[6] [Dio99] versions of the translinear loop multiplier covered in section 2.4.2. The output of both multipliers is subtracted, in order to avoid detecting purely temporal variations such as flicker [Har98b].

To test the vision chip, it was mounted on a Koala robot, facing forward [Har98a][Tel98]. The EMD outputs were spatially integrated (summed) across the retina, obtaining a single voltage proportional to the average motion in the image. This voltage was read through an analog input pin of the robot and the motors were commanded to rotate the robot in order to compensate the visual signal. The robot was placed on a platform in front of an unconstrained office scene and the platform rotated. Up to moderate angular speeds, the robot was able to stabilize itself.

2.6.2 Edge-tracking retina

The circuit discussed in this example does not model a particular animal, but it uses some mechanisms that are inspired from biology. The purpose of the vision chip is to select a feature of interest and track it. Changes in contrast, i.e., edges, were selected as features to be considered, and the feature of interest is the one with the highest dark-to-bright or bright-to-dark variation.

The circuit structure, described in [Kra96], is shown in figure 2.24. As a first stage it uses Delbruck's *adaptive photoreceptors* [Del96] (cf. sec. 2.5.2), whose output is smoothed out by a resistive network (cf. sec 2.4.4). The second stage is an array of transconductance amplifiers (cf. sec. 2.4.2) that carry out a spatial derivative of the signal by subtracting each node from its neighbor. At the output of the amplifier array a series of gates afford the selection of rising edges (positive output current) or falling edges (negative output current) or both. Thus, edge polarity can be externally selected by opening the desired gates. The output currents are in turn fed into a Winner-take-all circuit (cf. sec. 2.4.5) that selects the edge (rising or falling) with the highest slope. This circuit is a hysteretic WTA [Ind97]; it has lateral resistive coupling between individual cells that allow the circuit to lock onto the highest input, even if this one is moving. The amount of locking can be externally controlled. The last stage is a *centroid circuit* [Dew92] that outputs a voltage proportional to the winner's position. The circuit consists of an array of voltage followers in parallel, where only one of them will be on at a given time (because of the WTA). Since it is an array of voltage followers, it will set V_{out} to the value of the input voltage V_i, which in turn is proportional to the distance to the extremities.

The edge tracking retina was mounted on a Koala robot, centered at the front with a 60^o depression so that it could sense the floor 10 cm in front of the robot [Kra96] [Tel98]. A black electrical cord lying on the floor was used to generate the line to be followed. The chip output voltage V_{out} was connected to one of the robot's analog inputs and converted with 8-bit resolution. The rotational component of the motor commands is proportional to the edge displacement from the center position, and the forward component has a Gaussian profile, with the maximum speed occurring when the edge is centered in the visual field. By using such a simple control algorithm, the robot was able to follow the black line in real time.

[6] FET: Field-Effect Transistor. See glossary.

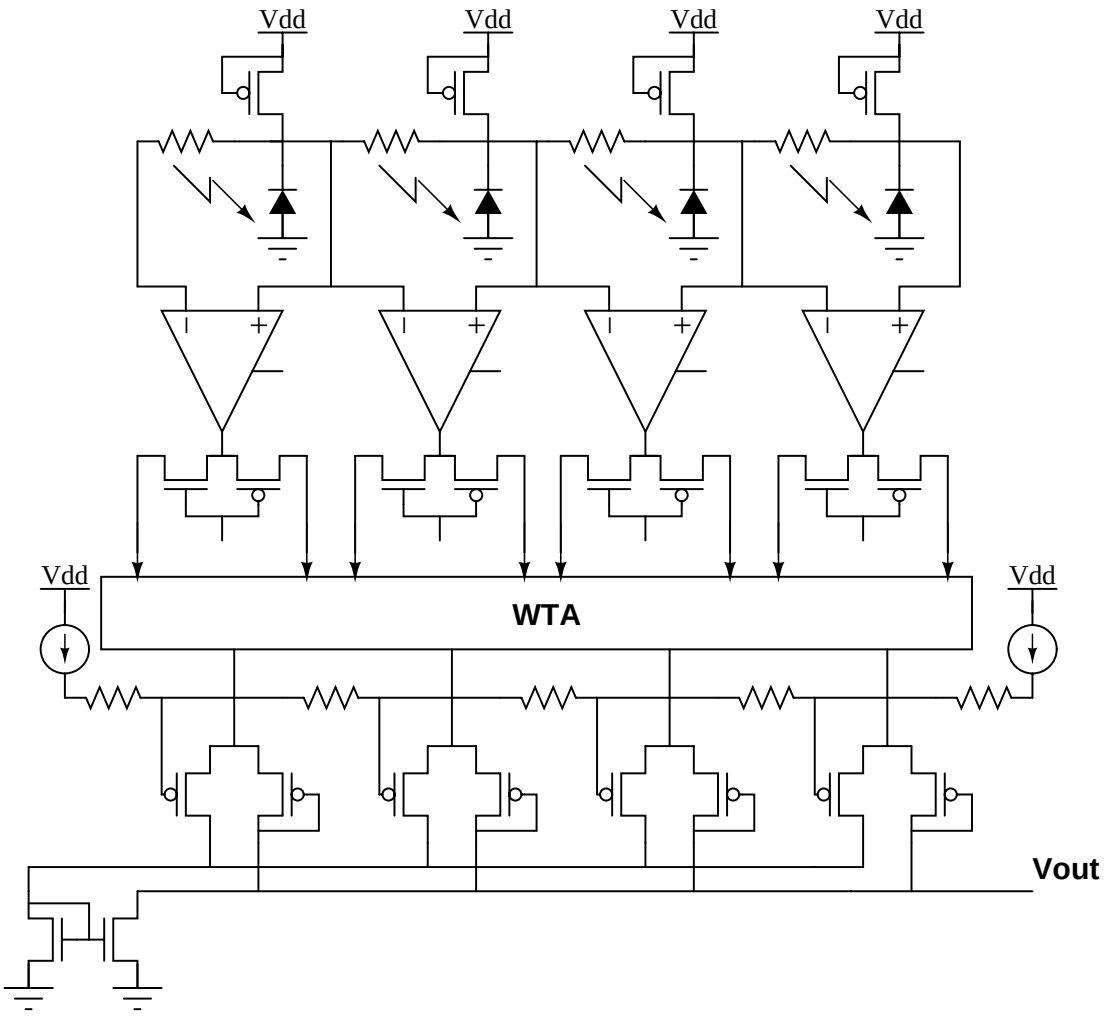


Figure 2.24: Block diagram of an A-VLSI tracking retina. V_{out} is proportional to the position of the strongest edge on the image. From [KRA96].

2.7 Summary

This chapter has discussed some advantages and inconvenients of analog VLSI circuits. Autonomous mobile robotics can benefit from the sensing and processing capabilities of A-VLSI. The key advantages for mobile robots are the sensors small size, integration, real-time processing capabilities and low power consumption. Neuromorphism is advantageous too, for it brings to mobile robots some principles that make biological systems so efficient.

Analog circuits have some drawbacks. For the end user, Reduced precision is the most important of them, but in most cases it is not a serious obstacle. On one hand perception systems do not require a high degree of precision, as communication or storage systems do. On the other hand, some coding techniques, such as place coding [LAN98] can improve precision. Other drawbacks of analog systems are lack of flexibility and difficult, almost handmade design.

Algorithms that can be decomposed into the parallel computation of numerous simple units with limited intercommunication are easier to map into a silicon substrate. Although the computational units are simple, highly complex and nonlinear behaviors can be obtained by establishing interactions among units and between units and common signals.

Design is based on several building blocks capable of computing linear and nonlinear operations. These building blocks have been discussed to introduce the reader to kind of operations that can be solved on analog VLSI, but the list is far from exhaustive. The building blocks were chosen for their relevance in image sensing and spatial processing. Other building blocks could be defined for time-domain processing, such as sound processing, or pulse-based coding.

Three examples of A-VLSI applications for mobile robotics are described. Except for Franceschini's project, robots were only used as a test bench or showcase for the analog retinas, and not as an integral part of the project. Analog VLSI circuits have yet to reach the point where they are developed as a solution to a mobile robotics problem, and not the other way around.

Chapter 3

Stereoscopic vision in animals and robots

Contents

3.1 Introduction

Depth perception is one of the most important visual processes in human and upper vertebrates, since it allows these animals to perform high-precision visuomotor tasks such as prey catching and grasping. Depth perception in animals is usually achieved by several visual mechanisms.

1. Accommodation of the optic system is one of them. For a given focal setup, given a short depth of field, objects at a given distance appear sharp and those in the foreground or the background appear fuzzy. This monotonic relationship between focus and distance is exploited to provide *depth from focus* cues.

2. Other cues that can be used extract *depth from motion*. The reader has certainly noticed, while traveling by train, that distant objects appear to be static, whereas nearby objects seem to move faster (and indeed their retinal projections do move faster.) Such an effect, called *motion parallax* can be exploited to obtain the object's distance. If the velocity is known, absolute distance can be calculated. If it is unknown, the ratio distance/velocity, known as *time-to-contact*, can be obtained. Such a scheme can be found in the fly's visual system [Fra92].

3. Cognitive cues are also known to be exploited, particularly in humans. A priori knowledge of an object's size can be used to calculate the distance based on the objects subtended angle. Knowledge of the objects' shape can be used to recover their relative depth based on *occlusions*, the partial covering of an object by another in the foreground.

4. The most common process for depth perception in upper vertebrates is binocular stereoscopy or *stereopsis*. Stereoscopy comes from the Greek words $\sigma\tau\epsilon\rho\epsilon\acute{o}\varsigma$[1], solid and $\sigma\kappa o\pi\acute{\epsilon}\iota\nu$, to view. It is therefore, the ability to see, to perceive solids, volumes. Stereopsis is the ability to measure distances based on the simultaneous perception of the same scene from two (or more) points of view. Distant objects have the same or very similar retinal projections in both eyes, whereas nearby objects have widely different retinal projections.

Each of these depth-perception mechanisms have their own particular drawbacks or limitations. Depth from focus implies a mobile optic system with a limited *depth of field*. Depth from motion not only requires constant motion in order to measure distances, but it also has a bothersome "blind region" straight ahead that must be compensated by some lateral motion.

[1] Not to be mistaken with $\sigma\tau\epsilon\rho\epsilon\acute{\upsilon}\omega$, to exhaust, to wear out.

Cognitive cues need a previous learning stage as well as an object recognition mechanism (not to mention a partially-occluded object recognition mechanism.) Finally, stereoscopy has a very complex mechanism to match object across images. Therefore, most animals seem to combine some of these mechanisms. Frogs use focal accommodation and stereopsis [Hou89], pigeons use stereopsis and motion parallax [Zei93] and human use stereopsis and cognitive cues [Chu92].

Method	Advantages	Disadvantages
Depth from focus	Monocular vision	Mobile optics, needs low depth of field
Depth from motion	Monocular vision	Need to know own speed, blind region
Cognitive cues	Monocular vision	Needs learning, lacks robustness
Stereoscopy	Robust, fully static	Binocular vision, computationally intensive

Table 3.1: Summary of depth-perception methods available in nature, with their advantages and disadvantages.

Nonetheless, the most powerful and most important depth-perception mechanism is stereoscopy, and that is the reason why this research project has focused on it. This chapter shall cover several aspects of stereoscopic vision. Section 3.2 will discuss the geometry of stereopsis and introduce several related terms that will appear later in this and the following chapters. Section 3.3 will cover some known facts about the structure and performance of stereoscopic vision in animals. The chapter will be closed with some examples of stereopsis applied to robots.

3.2 Some definitions on stereoscopy

This section starts with a geometrical description of stereoscopic vision that will lead to a statement of the stereopsis problem.

3.2.1 The epipolar constraint or, where did we loose the third dimension?

Suppose a binocular vision system consisting of two retinas or phototransducing arrays with their optic axes lying on the same plane and focusing on the fixation point F as in figure 3.1. For each point in the scene we would have 2 projections on the left and right retinas. It follows from figure 3.1 that any point lying in the segment $\overline{PF}$ in the scene will have its projection on segments $\overline{p_l f_l}$ and $\overline{p_r f_r}$ and any point in segment $\overline{PD}$ will be projected on segments $\overline{p_l d_l}$ and $\overline{p_r d_r}$.

The principle that information about one projected segment is related to information on an analogous projected segment is known as the *epipolar constraint*, and the two related segments are known as *epipolar lines*. Thus, to find the depth of segment $\overline{PF}$ we only need to consider the epipolar lines $\overline{p_l f_l}$ and $\overline{p_r f_r}$. By extension, the depth of any point lying on the plane defined by the optic axes (if they are rendered coplanar by fixating on a point F) can be extracted from the two horizontal epipolar lines, and the vertical projection can be ignored. If the point of interest is above or below the plane of the optic axes, as point D in figure 3.1, the corresponding epipolar lines will be slanted, but they can be determined by calibration [Aya89b][Kon98] and

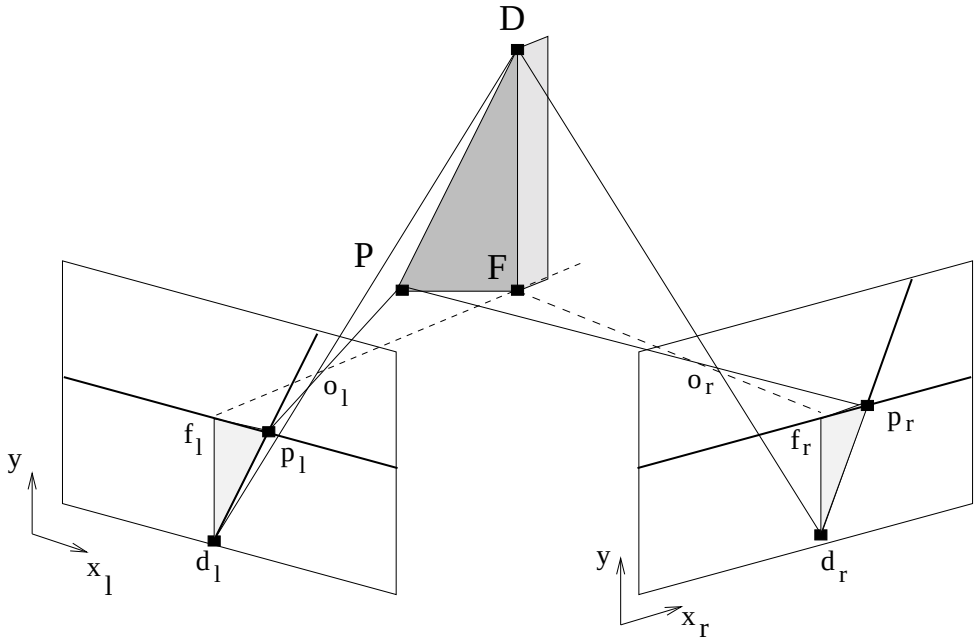


Figure 3.1: The epipolar lines for a given point in space are the projections of the planes formed by the optic axes and that point. To recover the depth coordinate of the point of interest only its two epipolar lines need to be considered. For instance, to measure the distance of point D, only the projected lines $\overline{f_l d_l}$ and $\overline{f_r d_r}$ need to be considered.

rotated into horizontal lines.

The problem of stereopsis can be treated as a 2D problem or a set of 2D problems [Pra85]. Hence, in this chapter and the following ones, only the information lying on horizontal, epipolar lines will be considered, ignoring the vertical coordinates of the projected scene. Such an approach is very common in robotic vision, because of the nature of the vision systems and the image contents. Indeed, for vision systems with parallel optic axes all epipolar lines are horizontal, and they coincide with the raster lines of CCD cameras [Rui96][Kon98]. Moreover, many robotics applications deal with indoor or office environments, where the vertical frequencies of the scene are very low, further justifying the use of a single horizontal line [Sri96].

3.2.2 The geometry of stereoscopy

If we consider the epipolar constraint on the setup in figure 3.1, we obtain a 2D projection that can be described by the diagram in figure 3.2. We have two eyes or vision sensors separated a distance B, called *baseline distance*. The optic axes, drawn in dashed lines, have a rotation of α and γ degrees around the y-axis, therefore crossing at a point F called *fixation point*. The angle $\phi = \alpha - \gamma$ is called *vergence angle*. A point P in the space in front of the eyes will be projected under an angle δ on the right eye and β on the left one. Angles δ and β are measured relative to the optic axes, increasing clockwise.

In the situation described in figure 3.2, the goal is to calculate the *depth* z and the *cyclopean*

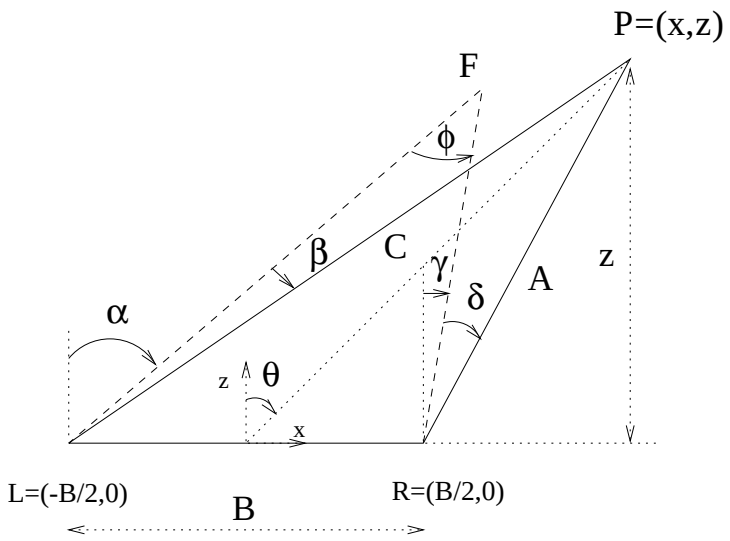


Figure 3.2: Geometric diagram of stereoscopic vision on the plane of the optic axes.

angle θ, the angle to an imaginary eye placed in the center of the baseline. We can define the equations for the lines of sight C and A:

$$z\cos(\frac{\pi}{2}-(\alpha+\beta))-(x+\frac{B}{2})\sin(\frac{\pi}{2}-(\alpha+\beta))=0 \Longrightarrow z\sin(\alpha+\beta)-(x+\frac{B}{2})\cos(\alpha+\beta)=0 \quad (3.1)$$

$$z\cos(\frac{\pi}{2}-(\gamma+\delta))-(x-\frac{B}{2})\sin(\frac{\pi}{2}-\gamma+\delta)=0 \Longrightarrow z\sin(\gamma+\delta)-(x-\frac{B}{2})\cos(\gamma+\delta)=0 \quad (3.2)$$

From eq. 3.1 we can isolate

$$z=(x+\frac{B}{2})\cdot\frac{1}{\tan(\alpha+\beta)} \quad (3.3)$$

and from eq. 3.2 we obtain

$$z=(x-\frac{B}{2})\cdot\frac{1}{\tan(\gamma+\delta)} \quad (3.4)$$

By replacing eq. 3.4 into eq. 3.3 we have

$$x=\frac{B}{2}\cdot\frac{\tan(\alpha+\beta)+\tan(\gamma+\delta)}{\tan(\alpha+\beta)-\tan(\gamma+\delta)} \quad (3.5)$$

which can be also written as

$$x=\frac{B}{2}\cdot\frac{\sin(\alpha+\beta+\gamma+\delta)}{\sin(\alpha+\beta-\gamma-\delta)} \quad (3.6)$$

Depth and cyclopean angle

We can replace equation 3.5 into eq. 3.4 to compute depth

$$z = \frac{B}{2}\left(1 + \frac{\tan(\alpha+\beta)+\tan(\gamma+\delta)}{\tan(\alpha+\beta)-\tan(\gamma+\delta)}\right)\cdot\frac{1}{\tan(\alpha+\beta)} = \frac{B}{2}\frac{2}{\tan(\alpha+\beta)-\tan(\gamma+\delta)} \Longrightarrow \quad (3.7)$$

$$z = \frac{B}{\tan(\alpha+\beta)-\tan(\gamma+\delta)} \quad (3.8)$$

For the particular case when the optic axes are parallel (i.e., there is no vergence), the previous equation simplifies to

$$z = \frac{B}{\tan(\beta)-\tan(\delta)} \quad (3.9)$$

The second variable of interest is the cyclopean angle

$$\tan(\theta) = \frac{x}{z} = \frac{\frac{B}{2}\cdot\frac{\tan(\alpha+\beta)+\tan(\gamma+\delta)}{\tan(\alpha+\beta)-\tan(\gamma+\delta)}}{\frac{B}{\tan(\alpha+\beta)-\tan(\gamma+\delta)}} \quad (3.10)$$

which simplifies to

$$\tan(\theta) = \frac{\tan(\alpha+\beta)+\tan(\gamma+\delta)}{2} \quad (3.11)$$

Most authors [Mah92][Mah89][Hen97a][Mar82] further simplify it by saying $\theta = \frac{\alpha+\beta+\gamma+\delta}{2}$ which is only true for frontal objects facing a vision system with small vergence, so that $\tan(\alpha+\beta) \approx \alpha+\beta$ and $\tan(\gamma+\delta) \approx \gamma+\delta$. While most of the aforementioned works did use a vision setup with small or nonexistent vergence, it is still a gross approximation for slightly peripheral objects. As it can be seen in figure 3.3, such an approximation reaches a 40 % error for objects lying at 40 degrees from the optic axes.

From equation 3.8 it follows that vision systems need to measure the angle under which they perceive the different points in the scene. Figure 3.4 shows the structure of a pinhole camera, consisting of a lens at a focal distance f from a focal plane, where the light transducers lie.

The angle β can be recovered from the position of the point's projection L_p on the focal plane, according to equation

$$\tan(\beta) = \frac{L_p}{f} \quad (3.12)$$

This result can be applied to eq. 3.9, yielding (for 2 cameras)

$$z = \frac{B}{\frac{L_p}{f}-\frac{R_p}{f}} = \frac{B\cdot f}{L_p - R_p} \quad (3.13)$$

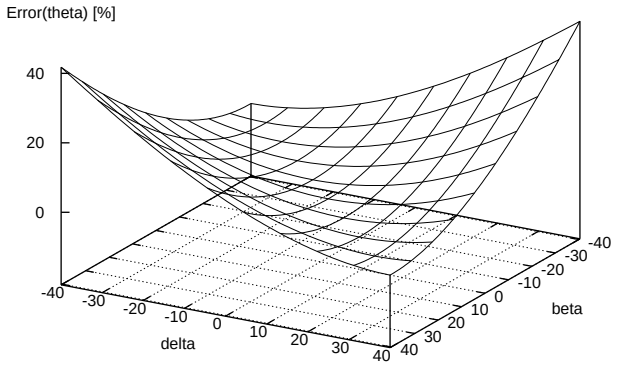


Figure 3.3: Error in the target's angular position due to the cyclopean angle approximation.

where L_p and R_p are the projections of point P on the left and right retinas respectively.

If the vision system has a small vergence and the objects lie at a small angle ($\tan(\beta) \approx \beta$), equation 3.12 is applied to the more general eq. 3.8, obtaining

$$z = \frac{B}{(\alpha + \frac{L_p}{f}) - (\gamma + \frac{R_p}{f})} = \frac{B \cdot f}{(L_p - R_p) + \phi \cdot f} \tag{3.14}$$

In case of higher vergence angle the relationship is nonlinear, but still analytically solvable.

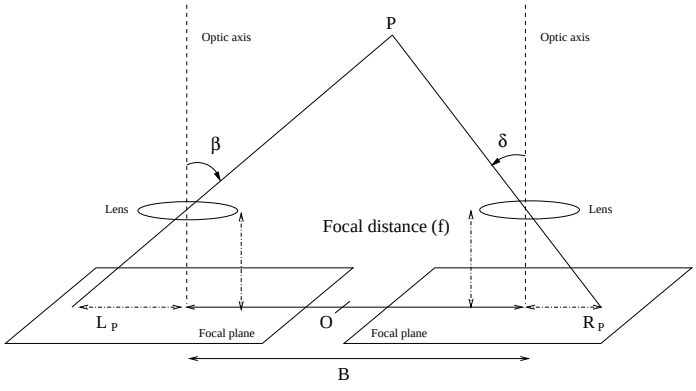


Figure 3.4: The pinhole camera: the angles β and δ under which point P is viewed can be calculated from its projections L_p and R_p and the focal distance f. The optic axes are parallel, making $\alpha = \gamma = 0$.

Disparity

Equation 3.13 tells us that depth z, i.e., the component of distance OP orthogonal to the baseline B (see figure 3.4), can be computed from the focal length f and the baseline distance B (two intrinsic parameters) and the relative difference in retinal coordinates $L_p - R_p$. We call *disparity* this relative difference of projections of any point P in the scene on the two eyes or cameras. If the baseline and focal length are known (they may be hardwired or learned during training or calibration), the *absolute depth* can be obtained. In the absence of this information, only a measure of *relative depth* can be recovered.

Horopter

In this second case, only 3 depths are considered: the points lying at the same depth as the fixation point (zero disparity), those lying behind it (*uncrossed disparity*) and those lying closer than the fixation point (*crossed disparity*) [MAR76]. The last two cases are sometimes referred as positive and negative disparities, but these terms should be avoided, since disparity can be evaluated as $L_p - R_p$ or $R_p - L_p$ depending on the author.

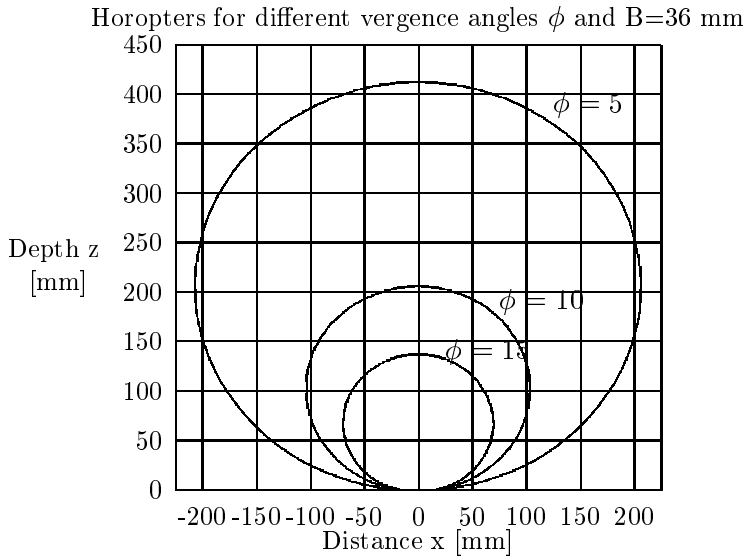


Figure 3.5: A stereoscopic vision system has the sensors at coordinates (-18,0,0) and (18,0,0). Its optic axes cross at a vergence angle ϕ on the horizontal plane $y = const$. The locus of points with disparity zero, called horopter, has a circular section on the plane of the optic axes. The figure shows the horopters for three possible vergence angles.

The locus of zero disparity is a 3D surface called *horopter*. That means that all points on the horopter have a constant relative difference in retinal projections equal to zero, or equivalently, that they are all seen under the same relative angle. On the plane formed by the optic axes, the section of the horopter is a circle, called *Vieth-Müller circle*, as shown in figure 3.5. The

derivation of the equations for the Vieth-Müller circle can be found in appendix A.

3.2.3 Stereopsis. Statement of the problem

From equation 3.13 it follows that the problem in stereoscopic vision can be decomposed in two steps or processes. The first one is the determination of the retinal projections of the points of interest in the scene, and the second one is the matching of the projections of the same point.

The first process is rather straightforward since most points in the 3D scene will have a projection on the retina. Nonetheless, a question arises about what are the most pertinent primitives that can yield projective invariants that will simplify the subsequent matching. Direct image intensity is not a good solution, since differences in lightning conditions or receptor sensitivity between left and right eyes could render the retinal projections difficult to match. Julesz's experiments [JUL71] with *random-dot-stereograms* (see figure 3.6) have shown that indeed humans are able to have depth perception in spite of illumination differences. Neurophysiological data have shown that oriented edges may be those invariants that produce a response in the cortex binocular neurons [PRA85]. It must also be mentioned that Julesz's experiments have demonstrated that object recognition is not necessary for stereoscopic perception, and indeed it seems that higher levels of processing are only applied on the combined image [CHU92][PET72].

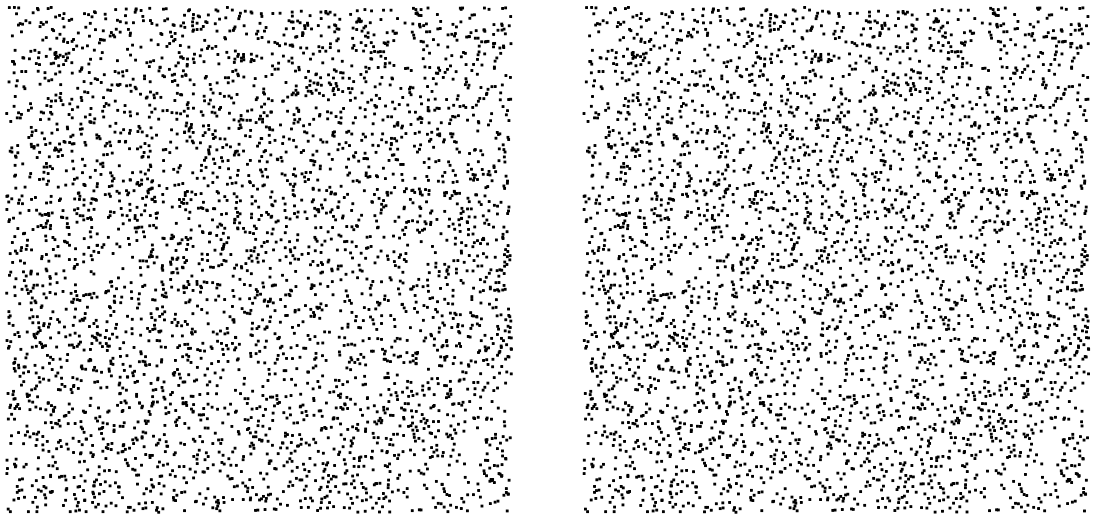


Figure 3.6: A random-dot stereogram. Place yourself in front of the image, at about 30 cm and try to fuse the two images into one by slightly squinting. A square area should raise above the background.

The second process, known as the *correspondence problem*, is by far the most difficult. If there are N feature points in the scene, N^2 possible matches will be generated and $\frac{(N^2)!}{N!(N^2-N)!}$ possible configurations will need to be considered. Most of these configurations will be discarded because they are geometrically impossible, but still, in some cases more than one valid solution can be found, as in figure 3.7. The task is further complicated by the presence of *occlusions*

that is, points in the scene that are only projected on one of the retinas, be it because they are partially hidden by an object, or because the noise on the retina prevents their detection. Target points that are correctly detected and matched can be *fused* to produce a single, cyclopean view of the scene.

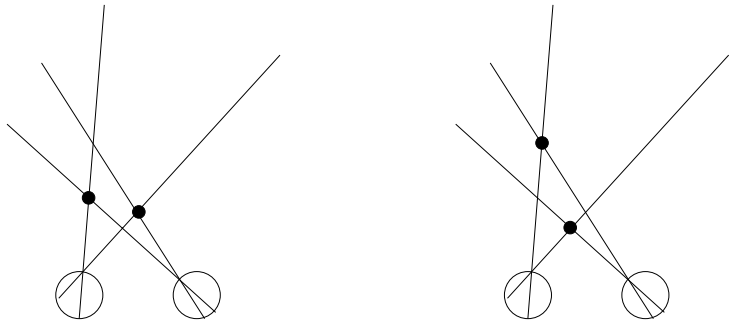


Figure 3.7: Two configurations in the real world lead to the same retinal projections. Which world are we really facing?

Unfortunately, only a partial knowledge of the functioning of stereoscopy has been gained, and it has lead to many different computational models, but there is not, up to date, a unified, complete explanation of this mechanism. The next two sections will cover some of these facts and discuss some computational models of particular relevance for mobile robots.

3.3 Stereoscopy in Nature

In the last twenty years robotics research has shown an interest in Nature as an inspiration for the robot structure and functionalities [And90] [Bro86] [Fra92]. In this section some facts of stereoscopy in animals and humans will be discussed. As mentioned in previous section, our knowledge about stereoscopic vision is very incomplete, and therefore a complete model can not be proposed for robotics applications. Nonetheless it can be interesting to know some facts and principles of stereopsis that may help roboticists in the design of their systems.

A second goal of this section is to show that in nature animals use stereoscopic vision in varying degrees, because their vision systems have evolved to meet the needs of their ecological niche. In general, herbivores typically have their eyes laterally placed, so as to have the widest field of view. They are thus able to detect predators coming from behind, but the small overlap between their fields of view precludes any depth perception from stereoscopy. On the other hand, most hunters, carnivores, have a good depth perception that allows them to detect their prey in spite of their camouflage techniques. Nonetheless, there are many variations inside these classes, and stereopsis should not be seen as a high level capacity of élite species [McF93]. Similarly, robots should show vision capabilities that are adapted to their tasks.

Two stereoscopic mechanisms have been distinguished in humans as well as animals [McF93]. Local stereopsis estimates the depth of unambiguous, matched points with respect to the fixation

point. It considers only 3 cases: the point lies at the same distance as the fixation point, behind it or in front of it [Chu92]. Global stereo is the mechanism whereby a quantitative measure of depth for each point in the scene is obtained, in spite of the ambiguities that may arise, as in Julesz's stereograms (Figure 3.6). To do so, global stereopsis has to consider global information coming from the whole image.

To explore the different structures and working principles, stereoscopy in humans will be described and compared it with that of animals.

3.3.1 Humans

Primates and humans possess the most efficient stereoscopic vision system in nature, and for obvious reasons, the most studied one, in particular from the psychophysics point of view. This section will overview the particularities of human stereopsis, from physiological to psychophysical results.

From eyes to brain: the optic chiasm

The images are formed in the retina, the phototransducing structure at the back of the eyeballs. For stereoscopic vision, the information coming from the *fovea*, the central, high-resolution, color-sensitive area [Bea94] at the center of the retina is the most important. Visual information leaves the eyes through the optic nerve, which conveys it to the brain.

For the brain to process the binocular information, the output from each eye has to be combined at a certain point. Anatomical analysis had shown that the optic nerves cross at the *optic chiasm* before entering the brain proper (see figure 3.8), but it was only in 1682 that Newton correctly hypothesized the presence of this exchange of information at the chiasm [Pet72]. Newton's concept of *partial decussation*, which was only verified last century, states that only some percentage of the nerve fibers cross at the chiasm, so that retinal information corresponding to the projection of the same area of the 3D scene enters the brain together. Thus, the fibers corresponding to nasal region of one retina are crossed (they are said to be *contralateral* fibers) to the other hemisphere, to join the *ipsilateral* fibers coming from the temporal[2] region of the other retina [Bus92].

The ratio of ipsilateral to contralateral fibers is proportional to the overlap between the retinal visual fields, reaching the value of 50 % for humans, slightly less for primates and decreasing sharply for other mammals.

Cortical organization

As shown in figure 3.8, the first waypoint for visual information in the brain is the *Lateral geniculate nucleus (LGN)*. In spite of the availability of information from both eyes, inputs coming from different eyes are not combined in any way. Cells in the LGN respond exclusively to monocular stimulus, and they are separated in layers according to the origin of the stimulus.

[2] As opposed to nasal.

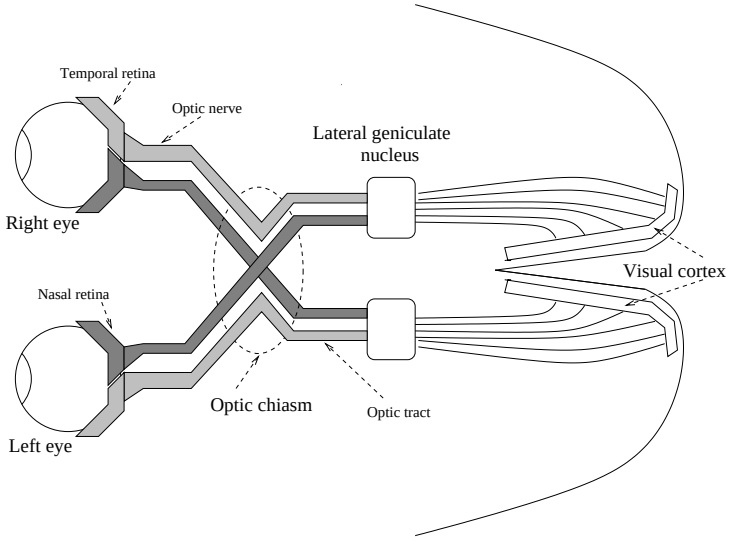


Figure 3.8: Schematic view of the stereoscopic visual pathway in the human brain. Feedback connections are not shown.

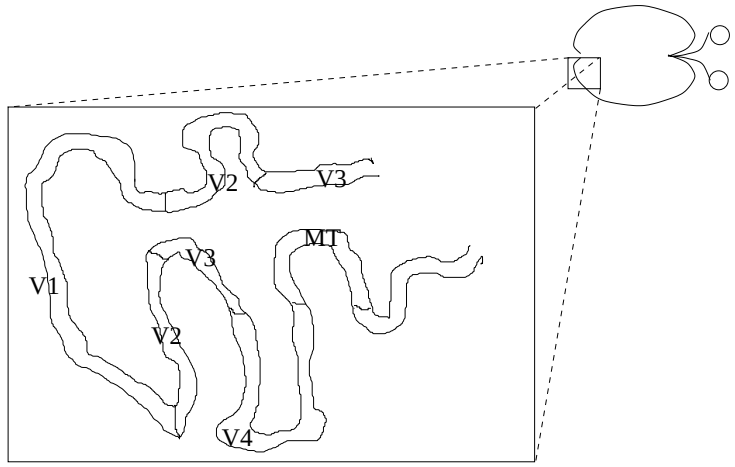


Figure 3.9: The visual cortex is divided into areas V1, V2, V3, V4 and MT.

Figure 3.10: A section of the brain showing the optic chiasm, the white spot in the center where the optic nerves cross. Taken from [LSP98]

The area of the brain responsible for visual processing, and hence stereoscopy, is the *visual cortex*, which lies at the back of the brain, as shown in diagram 3.8. The cortex is divided in several areas, called *V1* (primary visual cortex or striate cortex), V2, V3, V4 and MT (middle temporal area). See figure 3.9.

Area V1 is the entry-point of the visual cortex and together with V2 is the most directly related to stereoscopic processing. "Simple" cortical cells in area V1 of the cortex respond to stimuli with a particular orientation and polarity (eg. a dark to bright transition at 45^{o}). These simple cells are organized in *ocular dominance columns*, so that columns with monocular receptive fields (i.e., they respond to input from mostly one eye) alternate and are separated by columns responding to both eyes [Hub94] (c.f. figure 3.11). Each column in turn, groups the neurons responding to the same orientation, and some neurons respond only for targets at a given retinal position [Pet72]. "Complex" cells combine the signals from the neurons in each column and convey it to other cortical areas for further processing and possibly for fusion with other clues. The picture is of course overly simplified, because the processing is not strictly feed-forward. At all these levels feed-back connections are running and for instance, there are ten times more feedback connections from the cortex to the LGN than feed-forward ones [Chu92].

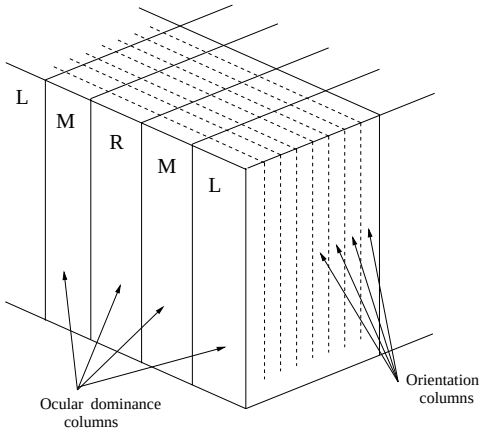


Figure 3.11: Diagram of the ocular dominance columns in the visual cortex (V1 area).

This topological organization simplifies the task of comparing inputs from both eyes, and indeed cells tuned to a given depth (i.e., disparity) have been detected in the V1 and V2 areas of visual cortex. Four classes of such neurons have been found, according to their response. *tuned cells* respond maximally to stimuli at the point of fixation, and their response decays very sharply when disparity is not zero. *Near cells* respond to stimuli lying in front of the horopter or surface of zero disparity, and *far cells* are activated by uncrossed stimuli. The fourth type of cells, called *disparity flat cells* seem to respond with identical strength across a wide range of disparities [POG84].

The limits of stereopsis

Artificial and natural stereoscopic vision systems show some limits to their depth-resolving capabilities. Some of these limiting factors arise from the geometry of stereoscopy, while others depend on the physical substrate of the photodetectors and the neural structure subserving the mechanism of fusion, and they are reflected on psychophysical experiments. In the following paragraphs some of these limits will be overviewed , since most of them also apply to robotic stereovision.

Panum's fusional area. Psychophysical experiments have shown that humans can only have stereoscopic perception in a small area around the horopter. This area is called *Panum's fusional area* and it has a variable depth of 0.16 to 0.33 degrees. Therefore our perception of a full-depth 3D world can only be obtained by scanning the scene, changing the fixation point. In humans, this scanning takes place every 200 ms [CHU92].

Stereoacuity. *acuity* is the reciprocal of the smallest resolvable distance between two different, spatially separated points in the scene, measured in either angle units or retinal coordinate units. It is the accuracy in spatial position perception [KRO86]. Acuity is of course dependent on the minimum distance between photoreceptors, that is, resolution. However psychophysical experiments, such as aligning two thin lines on a plane, measuring 1D acuity (called *Vernier*

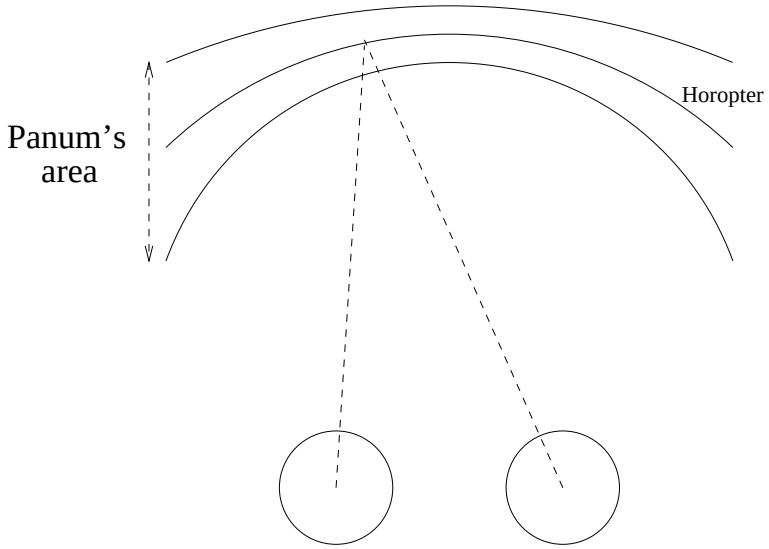


Figure 3.12: Panum's fusional area: stereopsis can only be achieved on a narrow area around the horopter.

acuity) show that humans can resolve a difference of 0.016^o [KRO86].

Stereoacuity is the measure of the smallest resolvable depth, and it can be calculated with a similar psychophysical experiment. In humans, it is estimated to be around $500 \cdot 10^{-6}$ degrees (2 seconds of arc), way below the intercone distance in the fovea, which is at least $5.5 \cdot 10^{-3}$ degrees. Such a condition, where acuity surpasses sensor resolution, is called *hyperacuity* [NAK85]. The mechanism of stereo hyperacuity appears to be based on *place coding* techniques [LAN98] applied on a population of near, far and tuned disparity cells. The resulting disparity is calculated as the average disparity of the excited cells, weighted by their intensity, and therefore it can reach any intermediate value. In this way, a subpixel accuracy can be obtained.

Disparity gradient. In most biological (and some artificial) systems, when two points with sharply different depth (and hence disparity) have very close cyclopean angles their depth position may not be correctly resolved. The system is then said to tolerate a maximum *disparity gradient* $\frac{\partial z}{\partial \theta}$ (see figure 3.2). In humans the disparity gradient is around one [MAH92].

3.3.2 Stereoscopy in vertebrates

Vertebrates, as humans, have two eyes, but their stereoscopic capabilities vary from species to species. For one thing, some animals have their eyes placed more laterally that others, favoring panoramic view to binocular view. Also, not all of them are capable of moving their eyes as it would be needed to fixate on different points. Another important factor is the structure of the retina, which may not always be optimized for stereoscopic vision. Birds retinas in particular present the greatest variability, including double foveated retinas and zooming mechanisms [ZEI93] [WAL90]. Finally, the nervous systems differ too, and some animals such as the frog

[HOU89], incapable of stereoscopic fusion, see double images but yet perceive depth.

Therefore, it must be stressed that binocular vision and stereopsis are not unitary phenomena [MCF93]. Stereopsis does require binocular vision, but binocular vision does not imply stereopsis.

Fish, reptiles and amphibians

In fish and reptiles, the chiasm has only contralateral connections, that is, all the fibers of the optic nerves cross. Furthermore, their eyes are either immobile or they can be moved independently, making it impossible to define a degree of binocular overlap [BUS92].

Frogs also have exclusively contralateral connections in the chiasm, but ipsilateral connections exist in the thalamus and corpus geniculatum, since binocular neurons have been detected there [HOU89]. Frogs do not have vergence movements, but they do show a 60^o binocular overlap, which combined with broad disparity-tuned cells in tectum and thalamus afford them to use binocular cues to catch prey and detect obstacles.

Birds

In birds, the chiasm is also completely crossed [BUS92], but they have a binocular overlap of their fields of view and stereopsis has been demonstrated behaviorally in some birds [CAS93] [MCF93]. The binocular overlap can only be studied in resting position, because many species have eye movements, and even nonconjugate eye movements, that change the effective visual overlap [MAR93]. According to this binocular field, birds can be classified into three groups:

- **Granivores and insectivores** are characterized by their narrow heads and lateral eyes, as in pigeons, domestic fowl and sparrows. They have a wide field of view in each eye, around 165^o but their optic axes are separated by approximately 165^o , thus yielding a binocular overlap of slightly less than 30^o [BUS92].

  In pigeons, stereoacuity for local stereopsis (without object recognition) is around 0.016^o , which is approximately the monocular acuity, approaching the hyperacuity range [MCF93]. Global stereoscopy however, has a stereoacuity of 0.158^o .

- **Diurnal predators** such as swallows, falcons and eagles, have around 140^o of monocular field of view, placed with their optic axes at 90^o of each other, and therefore these birds can reach 40 to 60 degrees of binocular overlap.

- **Nocturnal predators** such as owls, have immobile eyes, with a smaller field of view (approx. 70^o). Their optic axes are separated 50^o , producing a 60^o binocular overlap.

A comparison of binocular capabilities of different animals can be seen in table 3.2 in page 53.

Mammals

The same way as birds, mammals also present various degrees of stereoscopic capabilities. However, contrary to birds, they have more complex eye movements and structure. Scanning movements are conjugate, and therefore the angle between optic axes can be maintained. The second

Animal	Cyclopean field of view	Binocular overlap	Ipsilateral fibers
Pigeon	320°	30°	0 %
Eagle	220°	40-60°	0 %
Owl	160°	60°	0 %
Rabbit	360°	10-20°	10 %
Horse	360°	65°	20 %
Cat	190°	100°	40 %
Human	200°	140°	50 %

Table 3.2: Variation in binocular capabilities in animals. From [Bus92]

movement is that of vergence, that allows the optic axes to cross at a fixation point, but in turn complicates the measurement of the binocular field. The second characteristic that separates them from lower vertebrates is the partial decussation of the retinal projections at the optic chiasm.

Rodents (mouse, rat) and lagomorphs (rabbit, hare) have lateral eyes, with a wide field of view. For instance, rabbits have eyes of 190° of monocular view, separated 160° from each other, producing a cyclopean view of 360° and a binocular overlap of 15° . Their optic chiasm shows 10% of ipsilateral connections [Bus92].

Cats have frontal eyes with almost no divergence (≈7°) and 143° of monocular view angle. As a result, cats have 186° of cyclopean view and 98° of binocular overlap. They have a lower degree of decussation, with 40% of the optic nerve fibers having ipsilateral connections.

A general pattern can be concluded from these data. Prey, herbivorous animals such as the rabbit have developed a vision system that covers all 360° , allowing them to detect the approach of their natural enemies. On the other hand, predators such as cats and primates have evolved a vision system with a larger binocular view that allows them to detect a camouflaged prey [Pet90] and to display a very skilled visuomotor aiming [Bus92].

Figure 3.13: Only stereopsis can help a cheetah to pick out a zebra's neck out there.

3.4 Robot stereoscopy

Stereoscopic vision has been a subject of special interest for mobile robotics from almost its inception. As in biological systems, several methods are available for robots to perceive depth: motion parallax, range finding, depth-from-focus and stereoscopy, but stereo has important advantages. Contrary to motion parallax, it can yield a depth information even when the robot is static. It provides a measure of depth over a broader view than depth-from-focus, and it provides a richer, albeit less accurate information than range-finding devices, because the resulting cyclopean image can be further processed to obtain, for instance, object recognition.

Thus, many stereoscopic algorithms have been proposed in computer vision and robotics literature. This section will attempt to classify them and discuss the operation, results, advantages and inconvenients of the most important ones.

3.4.1 Constraints

As stated in section 3.2.3, solving the correspondence problem implies considering many possible matches, and only a small part of them will be correct. Researchers in computer vision have therefore tried to find some additional constraints to reduce the search space.

- *Epipolar constraint*: See section 3.2.1.

- *Compatibility constraint* [Mah92]. The strongest and the most self-evident of the constraints, it states that only similar features can be matched across images. Thus positive (raising) edges can only match positive edges, because edge polarity is a projection invariant.

- *Continuity constraint* [Mar82]. Objects in the scene are continuous, and disparity in the depth image can only vary smoothly, except at the object boundaries. The strict application of this principle favors solutions based on frontoparallel surfaces that is, surfaces parallel to the baseline.

  A serious criticism of this constraint can be made. The most common features used in related literature [Rui96] [Kon98] [Mah89] are edges, i.e., changes in pixel contrast, which mostly appear at object boundaries. Therefore, the very features we are manipulating are located at sharp disparity changes.

  A weaker version of this constrain has been proposed by Prazdny under the name of *coherence principle* [Pra85] to account for transparent surfaces where disparity can change. He states that "*a discontinuous disparity field may be a superposition of several interlaced continuous disparity fields each corresponding to a piecewise smooth surface.*"

- *Uniqueness constraint* [Aya89b]. Each feature in the monocular images can be assigned at most one disparity. In other words, a feature in one image can be matched with one and only one feature in the other image. This constraint is in contradiction with transparent surfaces in the scene, as well as at occlusion points, where two points in the scene may project on the same retinal position.

- *Ordering constraint* [Rui96]. In systems with parallel optic axes the horopter finds itself at infinity, and therefore all disparities will be crossed. This means that any point in the scene lying left of a given point will be projected on both retinas left of the given point's projections.

- *Symmetry constraint.* If feature i on the left image best matches with feature j on right image, then feature j should match feature i. In other words, the algorithm should be symmetric, and the solution should be the same (to a sign reversal) if the left and right images are swapped.

  This definition that I put forth has never been formally stated under such a name, but it is widely used in stereoscopic algorithms. In [Kon98] and [Pra85] those matches that do not comply with the constraint are deemed ambiguous and either eliminated or sent to a further processing stage. In [NAM95] it is used to calculate a confidence value for each pixel.

3.4.2 Classification

From an engineering standpoint, stereoscopy can be decomposed in two separate problems: *feature extraction* and *stereo correspondence*. The structure of the human brain seems to indicate that these problems are separately treated in the brain. But the existence of numerous feedback connections between different areas in the cortex may imply that these steps are not independent. Despite these facts, stereoscopic systems in computer vision and robotics seem to consider these problems as separate and independent.

Hence in the section that follows, when discussing the classification of computational stereopsis algorithms, the focus will be brought on stereomatching algorithms and the discussion on feature extraction will be left aside, except on particular occasions.

Feature-based methods

Feature-based methods follow in the strictest sense the task decomposition given above. A first processing stage extracts projective invariants, that is, image characteristics that are constant across images. This simplifies the subsequent correspondence problem, because the number of features in an image is small compared to the image size.

There are many cues that can be used as features, but the most common ones in computer vision are edges, that is changes in color or light intensity. Several filters capable of detecting changes can be found in literature. Examples include the Sobel filter [Mey98], Canny-Deriche [Rui96], time derivative [Mah89], *Laplacian of Gaussian* (LOG) [Kon98] and Gabor logons [Pöl96].

According to the type of filters in use, their number and the presence of any thresholding stages, features can be numeric (i.e., binary, scalar or vector) or symbolic. In the first case, algorithms found in computer vision and robotic vision literature can be roughly classified into:

1. Correlation-based stereo

A measure of distance in feature space is defined, so that the correspondence between two features can be numerically evaluated. Next, the feature images are correlated by using the feature distance: closer features receive a higher correlation value. Of course this yields all possible feature combinations.

By applying a threshold on the correlated image, the solution formed by the matches of highest correlation is obtained. Such an approach has been taken in [Mey98] and [Kon98]. Because of its interest for mobile robotics, the latter will be described in page 58. These correlation-based techniques are reliable only if features are robustly detected and they are easy to discriminate from each other.

If it is not the case, some of the constraints in section 3.4.1 must be applied. Prazdny thus applies the continuity constraint to add support to those matches whose close neighbors agree in disparity [Pra85]. Each correlation unit i receives a support $s(i,j)$ from each other unit j that is inversely proportional to the distance $|i-j|$ and the disparity difference $|d_i - d_j|$. The activation of each match is

$$A(i,j) = Corr(Left(i), Right(j)) + \max_j(s(i,j)) \tag{3.15}$$

He further uses the symmetry constraint to eliminate ambiguous matches.

In [Hen97a], the uniqueness constraint is applied, and the disparity along each line of sight is computed as the average disparity of those units that concur on a certain range of values.

Correlation-based stereoscopic algorithms are convenient because of their simplicity and noniterative computational structure. Nonetheless, they produce many false matches whose number can only be minimized by the clever application of some constraints. Other solutions include the addition of contextual information, as in [Kon98] and [Pöl96], which look for matches in 2D windows instead of isolated epipolar lines.

2. Iterative stereo

   Iterative stereomatching algorithms resemble the correlation-based ones. An initial guess is made by calculating the correlation between both feature images. This time however, a global "energy" evaluating the incoherence or entropy of the solution is defined. A suitable optimization algorithm is applied to iteratively eliminate false matches, bringing down the energy and reaching a (hopefully) stable incoherence minimum. Many optimization procedure can be used to do so.

   The Marr and Poggio [Mar76] is probably the most famous example of such an approach. It will be described in more detail in section 3.4.3. It is based on a lattice of correlators that interact among them through local inhibitory and excitatory connections that drive the system to a minimum energy state. Since a match is influenced by neighboring matches, it is also called *cooperative stereo*. Contrary to the previous type of algorithms, the neighbor's support or feedback is only local, and therefore several iterations must be

applied for the information to diffuse across all the lattice. Of course, in an analog hardware implementation of such algorithm the result would be obtained in real time.

Other optimization algorithms that have been successfully applied to iterative stereo include Hopfield neural networks [Rui96], simulated annealing [Bab95], cellular neural networks (CNN) [Zan98] and relaxation algorithms.

Iterative or cooperative stereo algorithms are robust to false matches and adequate for implementation on parallel hardware, both analog and digital. When implemented on sequential machines, they suffer from the long convergence time. Their most serious drawback, in serial and parallel implementation alike, is that the stability of the solution is not guaranteed.

3. Constructive stereo

   A particular variation of cooperative stereo is based on another optimization technique, dynamic programming [Oht85]. Under this paradigm, an initial match is selected (typically by applying the ordering constraint on the image borders) and the final solution is constructed by iteratively adding a new possible match to previous ones. Thus, instead of starting with all possible combinations and eliminating the false ones, these algorithms start from scratch and add new matches that keep the global incoherence or entropy low.

   These algorithms always reach stable and close to optimal solutions. Unfortunately it is difficult to pick the initial right match[3] and the algorithms cannot be parallelized or elegantly translated into hardware.

Symbolic matching. A special class of feature-based algorithms exists where the order between object recognition and stereocorrespondence is inverted. In lieu of matching simple features, these algorithms deal with higher-level features such as lines, junctions, and bends. Because of the nonquantitative nature of this information, I have called it *symbolic matching*. An example and its robotic application is described in page 59.

Intensity-based methods

Intensity-based methods avoid the problems of feature detection by trying to match image intensities directly. Of course, matching pixel by pixel would produce too many false matches, and therefore some other information has to be added. Contextual information, in the form of the neighboring pixels, is used to provide more discriminating information. Since in many cases the 2D neighbors are used, these methods are referred to as *area-based methods* [Hen97b].

Nonetheless, they can be interpreted as a particular case of correlation-based stereo algorithms with feature vectors, where the features are the light intensities on a neighborhood around the pixel of interest. It is a particular case where all pixels have an associated feature, and therefore a dense disparity map can be obtained.

[3] As it is often the case with the opposite sex.

As in correlation-based methods, a vector distance is defined. Absolute difference (L^1), square difference (L^2) or Hamming distance are the most common ones. Furthermore, stereo constraints (cf. section 3.4.1) are also applicable.

The main advantage of intensity-based methods is the dense disparity image that can be obtained. Moreover, by using interpolation techniques, disparity can be calculated to subpixel resolution. Of course, these two advantages imply a longer calculation time. Intensity-based methods are not resistant to illumination differences across images, which prevent their application in all but the most controlled situations. Furthermore, psychophysical experiments such as Julesz's show that the human stereovision system is not based on direct intensities [Jul71].

Phase-based stereo

In the absence of occlusions, the right image is a noisy, shifted version of the left image. The shift is the disparity that we are looking for. Phase-based maps compute disparity based on the shift theorem of the Fourier transform, which states that a shift in space (or time) domain produces a phase shift in the frequency domain [Opp89].

These methods therefore directly compute the disparity as the phase difference in the frequency spectrum of the left and right images. However the shift is not constant across the image, and therefore the Fourier transform, which yields a constant phase shift, cannot be applied. Instead, other frequency methods capable of computing a local shift are used. These include *Gabor functions* [NAM95] [Bis96], windowed Fourier transforms [Aya98], wavelets and Walsh filters [Adj96].

These methods have the advantage of producing stable solutions in a noniterative way and yielding dense disparity maps, because the phase can be calculated for each pixel in the images. Moreover they yield subpixel disparity resolution, since the phase difference need not be an integer. On the other hand, these methods can be computationally intensive, particularly when involving a direct Fourier transform.

These methods are of special interest for biologically-inspired stereo because there is evidence that neurons in the V1 area have receptive fields with a Gabor-like impulse response [Qia97b].

3.4.3 Examples

In order to show in detail the principles discussed above, some stereoscopic algorithms will be discussed. These algorithms have been chosen for their relevance in computational stereoscopy, their applicability on mobile robotic platforms and their suitability for VLSI implementation.

SRI's stereo board [Kon98]

The Stanford Research Institute's Small Vision Module (SVM) [Kon98] is a small stereo system made of commercially available components. It consists of two cameras and a processor that

implements SRI's area correlation algorithm.

After applying a calibration algorithm that calculates the epipolar lines, a Laplacian of Gaussian (LOG) filter is applied to both images. Such a filter will enhance the edges by subtracting the local average illumination, thus rendering the algorithm robust to intensity differences across images. The result of this processing can be interpreted as a feature vector made up by the 2D neighbors of a pixel, and therefore I have classified it as a feature-based, correlation kind of algorithm. In the next step these "features" are correlated across images by using the L^1 norm (absolute differences), to find the shift that produces the minimum distance. Search can be performed across 16, 24 or 32 disparities, and the result is interpolated by 4 to produce subpixel disparity resolution. The algorithm is further completed by two postprocessing stages. A "left/right check" [Kon98] is used to verify the symmetry constraint. Those matches that do not comply are identified as unmatched. The second stage uses an "interest operator" to eliminate matches in areas of little or no texture, where the resulting disparity may be driven by image noise. This interest operator requires a threshold that should be set at calibration time.

Two versions of the SVM exist. Both contain two CMOS 320 x 240 imagers (cf. section 2.5) with low-power A/D converters. The first version, called SVM+, has two *digital signal processors* (DSP) ADSP2181 at 33 MHz, and it has a processing rate of 2.5 frames per second. The system power consumption is estimated at 600 mW. The second version, SVM II, is based on a single TMS320C60x Texas Instruments DSP running at 200 MHz. This improved version can yield up to 30 frames per second.

Because of its size and relative low power, the SVM is very convenient for autonomous, mobile robotics and indeed it has already been successfully applied on a Pioneer I robot [Cha98].

Ayache's algorithm [Aya89b]

The algorithm described by Nicholas Ayache in [Aya89b] [Aya89a] is an example of a *symbolic matching* algorithm. He matches line segments across images. Segments are extracted from the images using the Canny-Deriche edge detector [Der87]. Next they are linked into polygons and stored as connected graphs that conserver the topological relationship among segments.

The stereomatching algorithm is of the constructive stereo kind starting with an initial *prediction*. This match produces a hypothesis *propagation* to neighboring segments in both graphs (symbolic images). A third step performs a *validation* of the previous hypothesis according to the epipolar, uniqueness, continuity and ordering constraints (cf. section 3.4.1. Some additional geometric constraints on the segment attributes (length, orientation) are also applied to eliminate false matches. By iteratively applying the prediction, propagation and validation steps, a solution for matching the left and right graphs is found.

The algorithm was tested on a stereo camera setup intended to be mounted on a mobile robot. The test scenes were static, indoor ones, taken in an office environment with a commercial CCD camera of 512 x 512 pixels. Preprocessing (segment extraction) by itself required a couple of minutes on a Sun 3 (1 MIPS) machine. The stereomatching algorithm required 6.7 s for two images/graphs of 385 and 401 segments on the same Sun 3 computer.

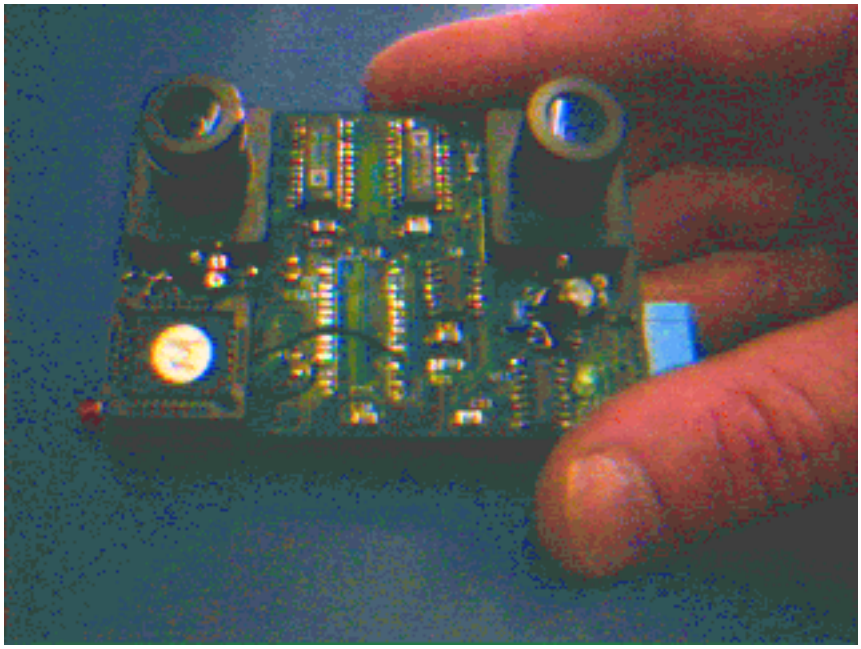


Figure 3.14: The SRI Small Vision Module. From [SVM].

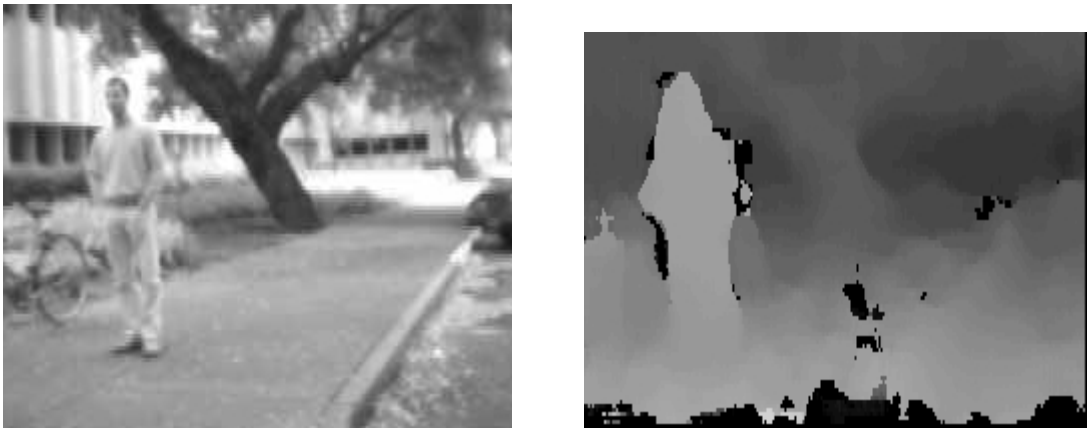


Figure 3.15: Example of the depth image obtained with the SVM. From [SVM].

Because of the high degree of discrimination provided by such symbolic features as segments are, this algorithm was able to yield depth maps with relatively few errors. Nonetheless, the time required for preprocessing and matching the images preclude it's use in mobile robots with small processing power.

Ruichek and Postaire's stereo algorithm [Rui96]

Within the framework of a European automated-driving research program, Yassine Ruichek and Gerard Postaire have developed a stereomatching algorithm for obstacle detection in front of an unmanned car. It is a feature-based algorithm of the iterative kind.

The algorithm, applied on linear images, begins by applying a Deriche edge detector [Der87] to extract the features in the image. As a part of this feature extraction, a small threshold t is applied to eliminate small intensity edges. Based on their sign (polarity) two classes are considered: raising and falling edges. Each class is separately matched by an iterative procedure, using a Hopfield neural network. Such a network consists of a lattice of neurons whose activation state u depends on an external input and the outputs of all the other neurons except itself according to the formula

$$\frac{u_{ik}}{dt} = -\frac{u_{ik}}{\tau} + \sum_j \sum_l W_{ikjl} E_{jl} + I_{ik} \tag{3.16}$$

where W_{ikjl} is the weight interconnecting neurons jl to ik, τ is a time constant, I_{ik} is the external driving input and E_{jl} is the output of the jl-th neuron, calculated as

$$E_{ik} = \frac{1}{2}(1 + \tanh(\frac{u_{ik}}{\lambda})) \tag{3.17}$$

A global energy of the network can be defined as

$$H = -\frac{1}{2} \sum_i \sum_k \sum_j \sum_l W_{ikjl} E_{ik} E_{jl} - \sum_i \sum_k I_{ik} E_{ik} \tag{3.18}$$

Ruichek et al. define the weights W and inputs I according to the the uniqueness, ordering and continuity constraints (cf. section 3.4.1) plus a geometric constraint, so that minimizing the energy H is equivalent to satisfying these constraints. Equation 3.16 is solved by the Runge-Kutta method, iteratively reducing the global energy H. The result is considered to have converged when there is no update in the neurons. At that point, the most active neurons per row and column define the correct matches.

Two linear CCD cameras of 2048 pixels each were placed on top of a car with parallel optic axes and slightly tilted down so that they intersected the road at about 20 m in front of the car. The algorithm was implemented on a PC equipped with a 486DX at 33 MHz and two extra i860 processors, obtaining a throughput of 10 frames per second. This allowed them to drive the car at 100 Km/h while detecting obstacles within a safe distance.

The algorithm provides a fairly good rate for real-time applications, although this has probably been favored by the relative absence of edges on the pavement. This method has the advantage of being highly parallelizable and suitable for digital hardware implementations. The

need for global interconnections make it very difficult to implement in A-VLSI. As with most feature-based systems, it has the drawback of producing sparse disparity maps, because the space in between features can not be interpolated.

Marr and Poggio's cooperative algorithm [MAR82] [MAH89]

In 1976 Marr and Poggio put forth a stereo matching algorithm based on "parallel, recurrent, nonlinear interactions, both excitatory and inhibitory" [MAR76]. It is therefore a feature-based, iterative algorithm.

The matching algorithm uses a square lattice of correlation cells, each receiving input from a left and a right retinal position. To impose the continuity constraint, cells have an excitatory connection from neighbors at the same disparity. The uniqueness constraint is realized in the form of inhibitory connections along the lines of sight. Figure 3.16 shows the structure of the net. The algorithm calls for an iterative calculation of the individual neuron's activation, until a stable state is reached which should be close to the correct solution.

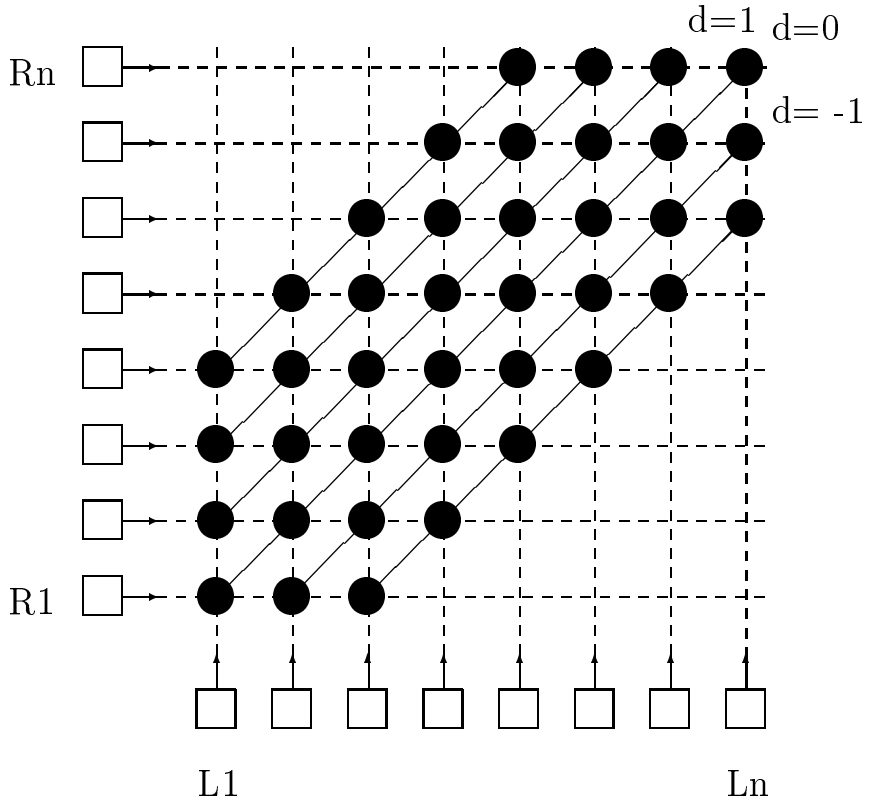


Figure 3.16: Lattice structure of the Marr and Poggio algorithm. Dashed lines indicate inhibitory connections along lines of sight and solid lines show excitatory connections across isodisparity surfaces.

Because of the local nature of the excitatory and inhibitory connections, this algorithm lend itself naturally for an analog VLSI implementation. In 1989, Mahowald and Delbrück pre-

sented such a chip [MAH89], and it is rather this hardware implementation that will be analyzed.

Local changes in contrast where selected as input features to the stereomatching array. To detect these changes, a high pass filter implemented as a resistive network was used, as described in page 26. Pixels brighter than their neighbors produce positive features and those pixels darker than the local average yield negative feature values. These features are in turn fed to the correlator array, a biased Gilbert multiplier that calculates the correlation

$$I_{corr} = \frac{I_b}{2}(1 + \tanh(V_r)\tanh(V_l)) \tag{3.19}$$

where V_r and V_l are the features expressed as differential voltages.

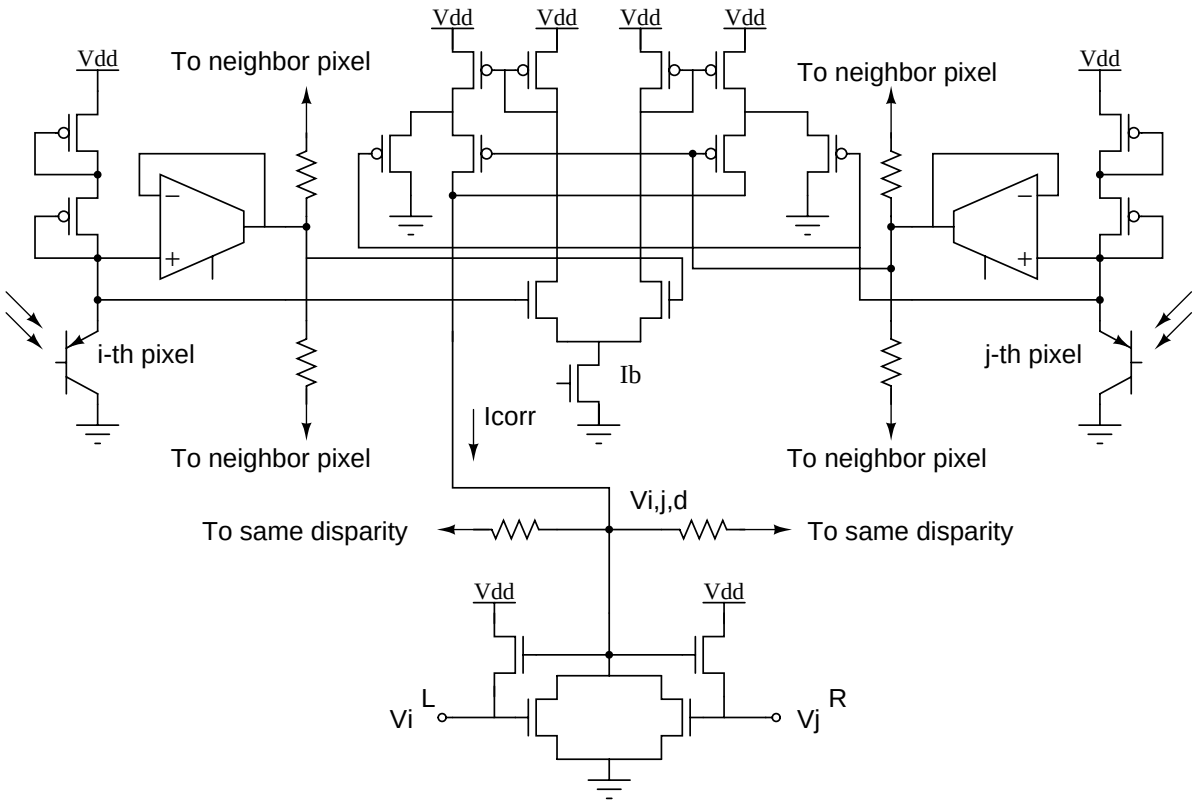


Figure 3.17: A single cell in Mahowald and Delbrück's implementation of Marr and Poggio's cooperative stereo algorithm.

It can be easily seen the correlator, shown in the center of figure 3.17, performs a kind of XOR functions, so that matches involving edges with the same polarity are assigned a value above $\frac{I_b}{2}$ and those involving opposite kind of features produce a low correlation current I_{corr}. This current is laterally diffused by means of a resistive network (the horizontal resistors in figure 3.17), thus producing an excitatory connection to neighboring cells lying at the same disparity line. The remaining current is used as input to two Winner-Take-All circuits (cf. section 2.4.5 that implement the inhibitory, competitive connection with other correlator cells lying

at the same lines of sight i and j. Voltages V_i^L and V_j^R correspond to the common voltages of the left line-of-sight and right line-of-sight WTA arrays. The output of the chip is taken at nodes V_{ijd}.

The resistors that control the coupling along disparity surfaces are variable ones. They help in suppressing the false matches and filling in the solutions in areas where the retinal input is too low to generate a match. Therefore, a dense depth map can be obtained if the coupling is moderate to high. If two correlators on the same WTA array produce a similar I_{corr} output both will be winners and the result can be interpreted as if they were lying at the average disparity of these units. Hence subpixel resolution is also available on this implementation.

Mahowald and Delbrück built a chip[4] on 2μm CMOS process, with two 40-pixel retinas lying parallel on the same surface, and the image focused by a single lens. Stereograms were used as stimuli. When the correlator inputs were too high or too low, driving the correlator to saturation many false matches occurred, because the WTA could not correctly discriminate them on their intensities. The same happens on the software version of the Marr and Poggio algorithm if binary (saturated) features are used. For normal inputs, the chip correctly interpreted the image, although a physical limitation was found whereby only 2 disparity surfaces could be discriminated.

Although the size, real-time processing and power consumption make it ideally suited for a mobile robotics application, the chip could not be tested on a robot because the baseline was too small and the single-lens setup prevented it from working on real images.

3.5 Summary

Stereoscopy is not the only method of obtaining a depth perception. Motion parallax and depth from focus are also used in natural and artificial vision systems. Nonetheless, stereopsis is the most advantageous one, for animals and robots alike. It allows predators to track their prey while staying motionless [Pet90]. The geometry of the stereoscopic problem shows us the extent of these advantages, as well as some limitations.

Knowledge of the stereopsis mechanism in biological systems is not complete enough to produce a model that could be experimentally or analytically validated. This has not prevented the computer vision researchers from producing several algorithms capable of solving the stereoscopic problem. Independently of their varying degrees of success when confronted with psychophysical and neurobiological experimental data, they do provide artificial vision systems with some depth perception.

[4] Actually two versions were built. The other version used a time derivative circuit to produce features, but the correlator structure was identical. [Mah89]

Part II

Stereo algorithms for small robots

Preamble

As it can be seen from the previous chapter, several stereoscopic algorithms and models have been proposed and tried in the field of computer vision and neurobiology. Each of them has advantages and inconvenients that make it more adequate for specific applications, and not a single algorithm can be said to correctly model the mechanism of biological stereovision nor to have a performance clearly above the others.

The main criterion for the choice of stereoscopic algorithms presented in this chapter is their applicability to small mobile robots. In particular, the attention has been borne on algorithms likely to be implemented in analog VLSI so as to fully take advantage of the benefits of this support, discussed in chapter 2. This implies some restrictions on the algorithms:

1. A certain degree of simplicity, since complex, nonlinear calculations are more difficult to map on silicon.
2. Memoryless computation. Any storage of intermediate values such as image buffers has been avoided. Although it is true that analog VLSI circuits can store values by using capacitors or better yet, floating-gate circuits [Dio99], the complexity of these circuits together with their limited precision warrant their exclusion at this time.
3. Local processing. In analog VLSI, because of its real-time characteristics, making information globally available requires a direct connection from each information source to each destination. This complicates the design (all these lines must be routed) and increases the costs by increasing the area. Therefore this research has focused on algorithms that only require local information, that is, signals from neighboring nodes, and a limited amount of information about the overall system state.
4. Parallelism. A corollary of the first and third restrictions. If computation is to be simple and local, parallelism that is, the use of multiple identical units is the best available solution to have the necessary total computational power.

An analog VLSI chip has not been built within the scope of this project, but the algorithms have been simulated and tested on digital systems, but always keeping in mind their likely implementation in A-VLSI. Digital systems benefit too from the constraints mentioned heretofore: memoryless computation reduces the amount of RAM needed, and simplicity allows for the use of simpler and faster microprocessors, or even Field Programmable Gate Arrays *(FPGAs)*.

Chapter 4 will present a phase-based algorithm based on Gábor functions and propose an implementation both in analog VLSI and digital systems, and show the results when applied to the Khepera robot. Chapter 5 covers a stereoscopic algorithm based on image differences, its analog VLSI implementation and experimental data.

Chapter 4

A stereoscopic algorithm based on Gábor functions

Contents

4.1 Background

The algorithm described in this chapter belongs to the class of phase-based stereovision algorithms. As discussed in section 3.4.2, these algorithms attempt to recover the *disparity* or spatial shift from the phase shift of the signals in the frequency domain.

The actual method that will be described is based on the phase properties of the *Gábor function*. A particular care must be taken with the use of the 'Gabor' keyword. Other stereoscopic algorithms may employ different properties of the Gábor function to measure the *depth* from a pair of images. For instance, the algorithm described in [PÖL96] utilizes the Gábor function as a feature extractor, while performing a correlation to match these features across images. Also [DUC97] [SME98] and [HEE88] show how the feature-detection properties of the Gábor function are employed. On the contrary, in the algorithm that will discussed hereafter, as in [NAM95], [QIA97A] and [FLE93], the phase is the property exploited for depth extraction.

4.1.1 Phase recovery

If the phase of a signal is to be measured, a frequency-domain description of this signal is to be used, since frequency is the derivative of the instantaneous phase. The most common frequency-domain description of a signal is its Fourier transform. The frequency transform of an image $r(x)$ (the signal) is defined as:

$$R(\omega) = \mathcal{F}\{r(x)\} = \int_{-\infty}^{\infty} r(x) e^{-j\omega x} \, \mathrm{d}x \tag{4.1}$$

where ω is the pulsation, related to frequency f by $\omega = 2\pi f$.

If $r(x)$ were the scene as it is perceived by the right camera then, in accordance with the discussion in chapter 3.2.2, in a simplified case (without any occlusions) the left image $l(x)$ would look like a shifted version of the right one:

$$l(x + d) = r(x) \tag{4.2}$$

with the constant shift d depending on the *baseline distance* and the depth of the object in the scene. From equation 4.1 it follows that the Fourier transform of this image would be

$$L(\omega) = \mathcal{F}\{l(x)\} = \mathcal{F}\{r(x + d)\} = e^{j\omega d} R(\omega) \tag{4.3}$$

The disparity d can be recovered from the phase difference between $L(\omega)$ and $R(\omega)$.

For the particular problem that interests us, that of disparity detection, the Fourier transform has a fundamental weakness: it furnishes a global description of the image. If the shift is not constant, i.e., we have $r(x) = l(x + d(x))$, the Fourier transform does not provide any information about where (in spatial domain) does the particular shift happen. It yields an average disparity, because the Fourier transform only deals with stationary signals. The Fourier transform has no spatial localization (the "where") and precise frequency-domain localization (the "what"). A signal as precisely defined in space as a pulse has a Fourier transform of the form $sinc(\omega)$ that spreads all over the frequency spectrum, as seen in figure 4.1. Inversely, a

constant signal has a Fourier transform $\delta(w)$ that is precisely located in the frequency domain.

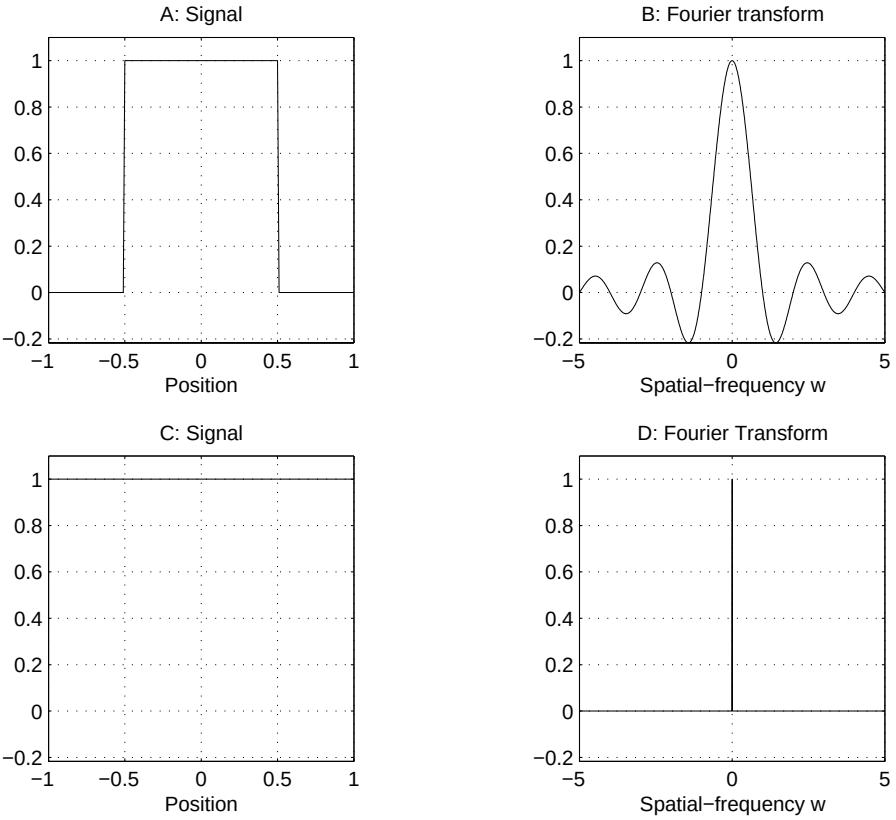


Figure 4.1: Heisenberg's Uncertainty Principle in signal processing: A signal well-localized (with limited spread) in the spatial domain (A) is completely delocalized in the frequency domain (B), and vice versa (C and D).

This signal-processing version of Heisenberg's Uncertainty Principle [GAB46] states that the signal length Δx and its frequency bandwidth $\Delta\omega$ follow the relation

$$\Delta x \cdot \Delta\omega \geq 0.5 \tag{4.4}$$

where the width and length are measured as the normalized variance of the energy distributions both in the spatial and spatial-frequency domains [BLO97]:

$$(\Delta x)^2 = \frac{\int_{-\infty}^{\infty} x^2 \|r(x)\|^2 \, \mathrm{d}x}{\int_{-\infty}^{\infty} \|r(x)\|^2 \, \mathrm{d}x} \tag{4.5}$$

$$(\Delta\omega)^2 = \frac{\int_{-\infty}^{\infty} \omega^2 \|R(\omega)\|^2 \, \mathrm{d}\omega}{\int_{-\infty}^{\infty} \|R(\omega)\|^2 \, \mathrm{d}\omega} \tag{4.6}$$

In terms of the recovery of depth from phase, Δw is the inverse of accuracy in the measurement of the frequency and hence of the phase[1] and therefore is inversely proportional to the

[1] frequency being the derivative of the instantaneous phase.

accuracy in depth perception. Correspondingly, Δx is the accuracy in determining the position of the point, that is, its *cyclopean angle* θ. Thus, for stereoscopic vision, equation 4.4 states

$$\Delta\theta \cdot \Delta z \geq k \tag{4.7}$$

where z is the depth as defined in page 40 and k is a positive constant.

4.1.2 Gábor function

The uncertainties discussed in the previous section can not be eliminated, but they can be minimized. In 1946 Dennis Gábor , 1971 Nobel Laureate for the development of holography, demonstrated that the lower uncertainty bound in equation 4.4 was reached for a complex function that has been since known as *Gábor function* [GAB46]. This function is a harmonic function that modulates a Gaussian envelope, and its 1-D version is

$$g(x) = e^{-\frac{x^2}{2\sigma^2}} \cdot e^{j\omega_o x} = gaus\beta_\sigma(x) \cdot e^{j\omega_o x} \tag{4.8}$$

where ω_o is the spatial pulsation (proportional to the spatial frequency) of the harmonic function and σ is the variance (spatial spread) of the Gaussian envelope. Its corresponding Fourier transform is a Gaussian

$$G(w) = \mathcal{F}\{g(x)\} = \sqrt{2\pi}\sigma\, e^{-\frac{\sigma^2(\omega-\omega_o)^2}{2}} = \sqrt{2\pi}\sigma\, gau\beta_{\frac{1}{\sigma}}(\omega - \omega_o) \tag{4.9}$$

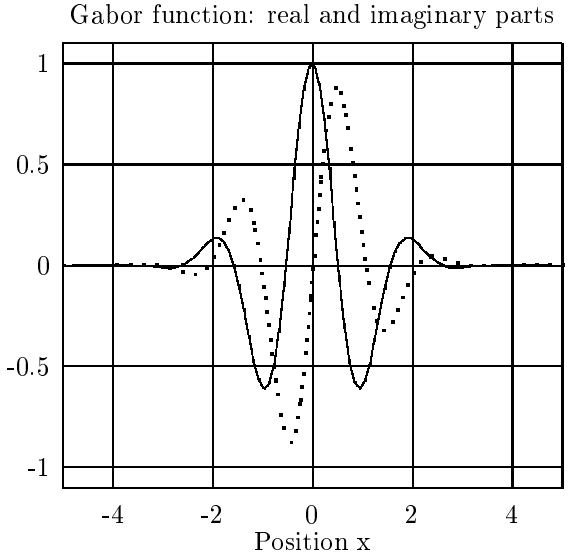


Figure 4.2: The real (solid) and imaginary (dotted) parts of the Gábor function, a Gaussian envelope modulated by a carrier.

Therefore, by using a kernel based on the Gábor function, a more accurate description of the signal, both in the spatial and spatial-frequency domains, can be obtained. To do so, it suffices to convolve the signal $s(x)$ with the Gábor function g_{ω_o} of pulsation ω_o and variance σ

$$
\begin{aligned}
s(x) * g_{\omega_o}(x) &= \int_{-\infty}^{\infty} s(\tau) g(x-\tau)\,\mathrm{d}\tau && (4.10)\\
&= \int_{-\infty}^{\infty} s(\tau) gau\beta(x-\tau) e^{j\omega_o(x-\tau)}\,\mathrm{d}\tau\\
&= \int_{-\infty}^{\infty} s(\tau) gau\beta(x-\tau) e^{j\omega_o x} e^{-j\omega_o \tau}\,\mathrm{d}\tau\\
&= e^{j\omega_o x} \int_{-\infty}^{\infty} s(\tau) gau\beta(x-\tau) e^{-j\omega_o \tau}\,\mathrm{d}\tau\\
&= e^{j\omega_o x} \int_{-\infty}^{\infty} h(\tau, x) e^{-j\omega_o \tau}\,\mathrm{d}\tau\\
&= e^{j\omega_o x} \mathcal{F}\{h(\tau, x)\}_{|\omega=\omega_o}\\
&= e^{j\omega_o x} H(\omega_o, x) && (4.11)
\end{aligned}
$$

where $h(\tau, x) \equiv s(\tau) gau\beta(x-\tau)$.

By applying the following properties of the Fourier transform

$$
\begin{aligned}
x(t)y(t) &\overset{\mathcal{F}}{\longleftrightarrow} \frac{1}{2\pi} X(\omega) * Y(\omega) && (4.12)\\
x(t-t_o) &\overset{\mathcal{F}}{\longleftrightarrow} X(\omega) e^{-j\omega t_o} && (4.13)\\
x(-t) &\overset{\mathcal{F}}{\longleftrightarrow} X(-\omega) && (4.14)\\
x(t)y(t-t_o) &\overset{\mathcal{F}}{\longleftrightarrow} X(\omega) * (Y(\omega) e^{j\omega t_o}) && (4.15)\\
gau\beta_\sigma(t) &\overset{\mathcal{F}}{\longleftrightarrow} \sqrt{2\pi}\sigma\, gau\beta_{\frac{1}{\sigma}}(\omega) && (4.16)
\end{aligned}
$$

the function $H(\omega_o, x)$ is found to be

$$
H(\omega_o, x) = \frac{\sigma}{\sqrt{2\pi}} S(\omega) * (gau\beta_{\frac{1}{\sigma}}(\omega) e^{j\omega x})_{|\omega=\omega_o} \qquad (4.17)
$$

and therefore, inserting equation 4.17 into eq. 4.10 we obtain

$$
s(x) * g_{\omega_o}(x) = \frac{\sigma}{\sqrt{2\pi}} e^{j\omega_o x} S(\omega) * (gau\beta_{\frac{1}{\sigma}}(\omega) e^{j\omega x})_{|\omega=\omega_o} \qquad (4.18)
$$

Interpretation. Eliminating the phase terms from equation 4.18, it can be seen that the convolution of a function $s(x)$ with a Gábor function of frequency ω_o is identical to the corresponding Fourier transform $S(\omega)$, smoothed out with a Gaussian function, and evaluated at the frequency $\omega = \omega_o$.

As discussed in the opening of section 4.1.1, a widening in the spatial-frequency domain, as it would be brought by a Gaussian smoothing, implies a thinning in the spatial domain, that is a more accurate localization in space.

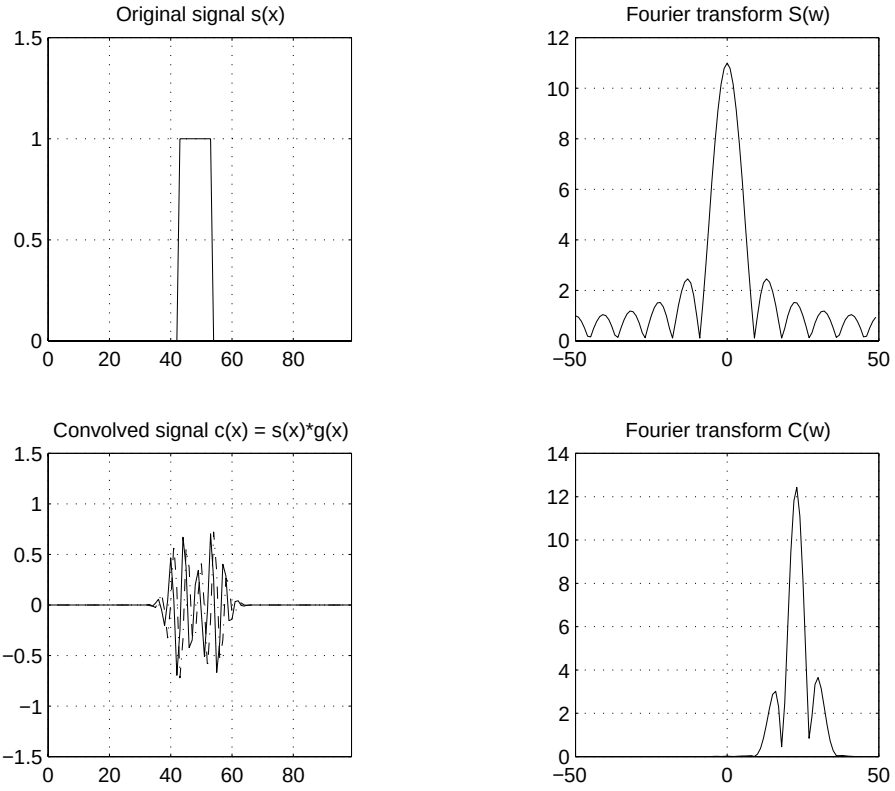


Figure 4.3: A signal (A), its Fourier transform (B), the same signal convolved with a Gábor function (C) and its Fourier transform (D). The effect of the Gábor function can be clearly seen: the space-domain signal is spread, reducing its localization, but in contrast, its spatial-frequency representation is narrowed, concentrating the energy of the signal in a smaller bandwidth.

4.2 Algorithm I

With this justification of the use of Gábor functions to measure the spectrum of a signal, we can set forth to recover the spectral phase of the two images $l(x)$ and $r(x)$, related by equation 4.19, as discussed in section 3.2.2.

$$r(x) = l(x + d(x)) \tag{4.19}$$

Because of the *epipolar constraint* (cf. section 3.2.1), the matching of two stereo images takes place on a line by line basis. For the calculations that follow, images will refer to linear or 1-D images, as in the raster lines on a screen.

By convolving the left and right images with a given Gábor function we obtain the complex functions

$$c_l(x) = l(x) * g(x)_{\omega_o,\sigma} = |c_l(x)|e^{j\varphi_l(x)} \tag{4.20}$$

$$c_r(x) = r(x) * g(x)_{\omega_o,\sigma} = |c_r(x)|e^{j\varphi_r(x)} \tag{4.21}$$

where $\varphi_l(x)$ and $\varphi_r(x)$ are the phases of complex functions $c_l(x)$ and $c_r(x)$ respectively.

By replacing equation 4.19 into 4.21 we have that

$$\begin{aligned} c_r(x) &= r(x) * g(x) = \int_{-\infty}^{\infty} r(t) e^{-\frac{(x-t)^2}{2\sigma^2}} e^{j\omega_o(x-t)}\,\mathrm{d}t \\ &= \int_{-\infty}^{\infty} l(t+d) e^{-\frac{(x-t)^2}{2\sigma^2}} e^{j\omega_o(x-t)}\,\mathrm{d}t \end{aligned} \tag{4.22}$$

A change of variables $t' = t + d$ yields

$$\begin{aligned} c_r(x) &= \int_{-\infty}^{\infty} l(t') e^{-\frac{(x-(t'-d))^2}{2\sigma^2}} e^{j\omega_o(x-t')} e^{j\omega_o d}\,\mathrm{d}t' \\ &= e^{j\omega_o d(x)} e^{-\frac{d^2}{2\sigma^2}} \int_{-\infty}^{\infty} l(t') e^{-\frac{(x-t')^2}{2\sigma^2}} e^{-\frac{2d(x-t')}{2\sigma^2}} e^{j\omega_o(x-t')}\,\mathrm{d}t' \end{aligned} \tag{4.23}$$

When the disparity d is smaller than the variance σ of the Gábor function, equation 4.23 can approximated by

$$c_r(x) \approx e^{j\omega_o d(x)} e^{-\frac{d^2}{2\sigma^2}} c_l(x) \tag{4.24}$$

Therefore the disparity $d(x)$ can be recovered from the phase difference, as

$$\omega_o d(x) = \varphi_r(x) - \varphi_l(x) \Rightarrow d(x) = \frac{\varphi_r(x) - \varphi_l(x)}{\omega_o} \tag{4.25}$$

Equation 4.25 shows some of the key properties and limitations of this algorithm:

1. $d(x)$ is continuous. This implies that the algorithm has stereo *hyperacuity*, i.e., its disparity resolution is higher than the individual retinal resolutions. If both resolutions were the same, only discrete values of $d(x)$ could be obtained, since the cameras are discrete.

2. $d(x)$ is defined for all x, although of course, not with the same degree of accuracy. Thus the algorithm produces a smooth disparity solution that may partially solve the *occlusion* problem.

3. There is a relationship between $d(x)$ and the modulating frequency ω_o. Indeed, the phase difference $\varphi_r(x) - \varphi_l(x)$ can only be in the $[-\pi, \pi)$ range, and therefore the range of detectable disparities, called *Panum's fusional area* is

$$d(x) \in [-\frac{\pi}{\omega_o}, \frac{\pi}{\omega_o}) \tag{4.26}$$

   When the optic axes are parallel, as it is the case in the vision systems that have been developed in this project (cf. chapter 7), disparity is always positive, and therefore the phase range can be shifted to $[0, 2\pi)$. In this case, the maximum disparity that can be recovered by a Gábor function of frequency ω_o is $\frac{2\pi}{\omega_o}$.

In both cases, the modulating frequency ω_o will be chosen so as to adjust Panum's area to the actual scene. Otherwise, because of the phase wrapping, points that are slightly closer than d_{max} will appear to be at the outer limit of the Panum's area.

4. The influence of σ does not appear explicitly in this equation, but it can be reasoned from equations 4.10, 4.20 and 4.21. Since the phase of c_l and c_r is compared for each pixel x in the image, and knowing that the pixel $l(x_o)$ does not necessarily correspond to $r(x_o)$ because of the disparity, it follows that the corresponding coefficients $c_l(x_o)$ and $c_r(x_o)$ must contain some information about neighboring pixels. The inclusion of neighboring pixels is done through a Gaussian windowing, as shown in eq. 4.10, and σ controls the spread of the Gaussian function.

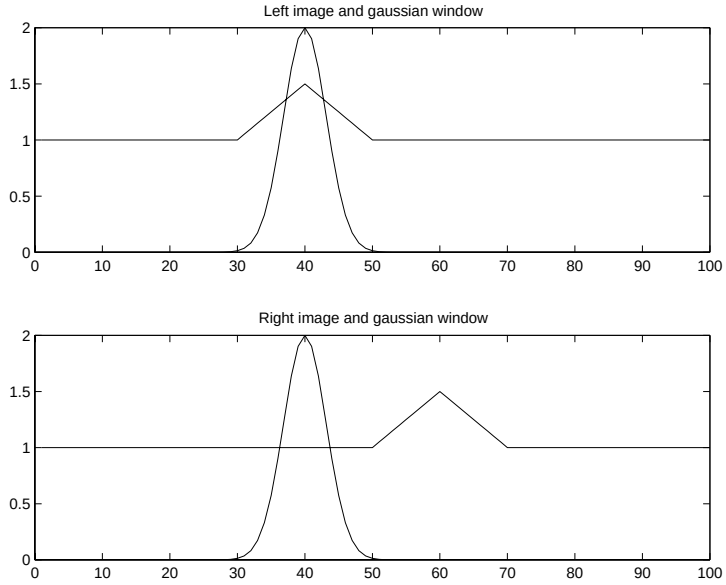


Figure 4.4: The influence of σ: If the Gaussian envelope of the Gábor function is too narrow, two corresponding values $c_l(40)$ and $c_r(40)$ (cf. eqs. 4.20 and 4.21) will not contain information about matching points in the image.

Therefore, to produce meaningful results the Gaussian window must contain the corresponding pixel, or in other words, the Gaussian width[2] must be more than twice the expected disparity.

5. There is not an explicit feature extraction stage. The Gábor filter is applied to all points in the image, irrespective of their saliency. There is however an implicit feature extraction, since the output functions c_l and c_r have their power concentrated around the edges in the image, as it can be perceived in figure 4.5. The algorithm based on the phase of Gábor functions can be said to favor edges, as the phase is difficult to determine at points of low power[3].

[2]The width of a Gaussian function can be considered to be 6σ, because at $gau\beta(3\sigma)$ the amplitude is approximately 1% of the maximum.

[3]Much as the direction of the wind is difficult to determine when it barely blows.

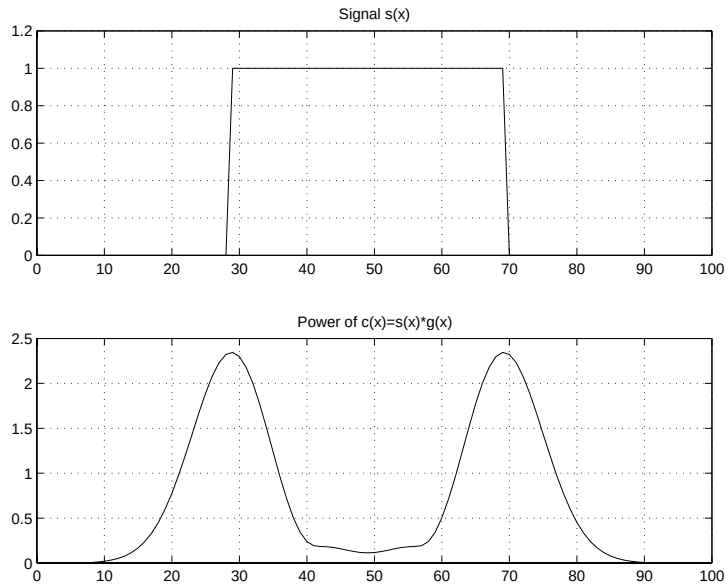


Figure 4.5: The power of the convolved signal $c(x) = s(x) * g(x)$ is concentrated around the edges of the original signal $s(x)$.

4.2.1 Confidence value

From the points 4 and 2 it can be understood that, even if σ is too small for the Gábor function to encompass the corresponding information on both images, the algorithm will still produce a given phase $\Delta\varphi(x)$ that will necessarily be false. Therefore a need for measure of the confidence in the algorithm's solution arises. Most researchers using Gábor functions [NAM95], [SAN88], [TRA98] use some kind of confidence value obtained as a function of the magnitudes of the left and right convolved images $|c_l(x)|$ and $|c_r|$.

Since ideally, that is if they were Fourier transforms, the convolved images $c_l(x)$ and $c_r(x)$ are supposed to contain the same information (i.e., magnitude) but with different phase (i.e., disparity), a situation where $|c_l(x_o)| \approx |c_r(x_o)|$ would indicate that indeed corresponding information has been matched and a high confidence value for that pixel x_o would be warranted. Similarly, if the magnitudes of $c_l(x_o)$ and $c_r(x_o)$ differ widely, a low confidence value can be assigned to the pixel.

For the experiments that follow, the confidence value is computed as in [SAN88], because of its simplicity, which enables its implementation in analog VLSI as well as better performance in digital systems. The confidence value $q(x)$ is set to vary between 0 (no confidence) and 1 (absolute confidence) by defining

$$q(x) = min(\frac{|c_l(x)|}{|c_r(x)|}, \frac{|c_r(x)|}{|c_l(x)|}) \in [0, 1] \tag{4.27}$$

4.2.2 Power threshold

Still, as indicated in point 5 in page 76, the algorithm will produce a phase $\varphi(x) = \angle c(x)$ for each pixel x. However, the phase can not be confidently determined when the magnitude $|c(x)|$ is almost zero. And yet, equation 4.27 would yield a high confidence value if both $|c_l(x)|$ and $|c_r(x)|$ are equal and close to zero.

To mitigate this problem, the actual value has to be taken into account. As in [NAM95] and [SAV99], a confidence threshold can be determined, based on the power $p(x)$

$$p^2(x) = |c_l(x)|^2 + |c_r(x)|^2 \tag{4.28}$$

If the power $p(x_i)$ of a certain pixel x_i in the image is below a certain level, the disparity measure at that point can be assigned a confidence level $q(x) = 0$.

4.2.3 Experiment I

Recapitulating, the algorithm described in the previous lines can be written as

Algorithm 4.1 Algorithm for the recovery of disparity d in the $left$ and $right$ images from their Gábor phase (Algorithm Gábor I)

```
[d] = gaborstereo(left,right)

  g = gabor(w,sigma);
  cl = convolution(left,g);          /* Eq. 4.20*/
  cr = convolution(right,g);         /* Eq. 4.21*/
  eps = 1e-6;
  qa = (eps+abs(cl))./(eps+abs(cr));
  qb = (eps+abs(cr))./(eps+abs(cl));
  q = min(qa,qb);                    /* Eq. 4.27*/
  fl = unwrap(angle(cl));            /* Eq. 4.20*/
  fr = unwrap(angle(cr));            /* Eq. 4.21*/
  dx = fr - fl;                      /* Eq. 4.25*/
  dx = dx + 2*pi*(dx<0);
  d = dx/omega;                      /* Eq. 4.25*/
  return d;
```

where `eps` is a small positive constant that is added to avoid a division by zero, `unwrap()` is a procedure to "unfold" the phase so that it runs in the $[0, \infty)$ range, instead of $[0, 2\pi)$, and a 2π correction is added to set the disparity to positive values only, which supposes that optic axes are parallel.

Test image

The first experiment was performed on a test stereoimage that has been synthetically generated. It consists of two `sinc()` functions with disparities 5 (leftmost peak) and 10 (rightmost peak).

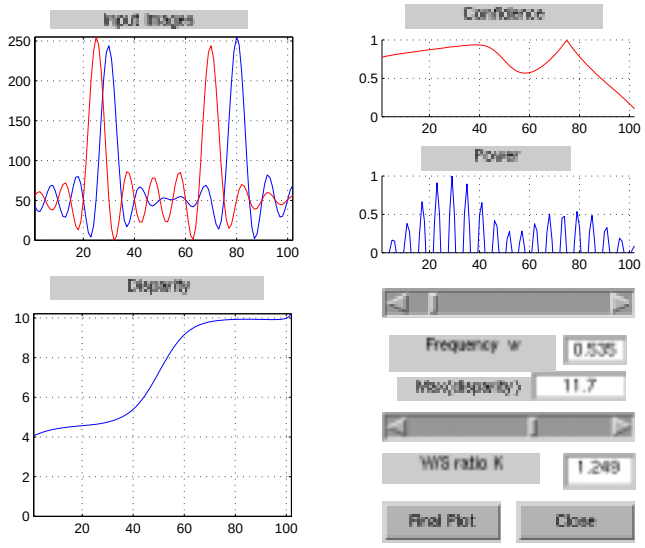


Figure 4.6: Stereomatching of two test images using the Gábor phase to recover disparity (algorithm I).

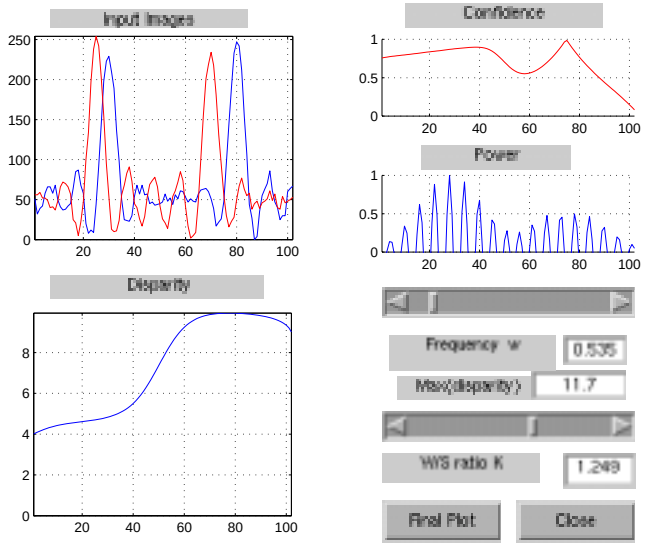


Figure 4.7: Stereomatching of two noisy test images (SNR = 20 dB) using the Gábor phase (algorithm I).

Figure 4.6 shows a screen shot of the `gabordemo` Matlab function that allows the user to interactively modify the Gábor function parameters to find a satisfying solution. The tunable parameters are the frequency ω_o and K, a constant relating frequency ω_o and deviation σ according to equation 4.29.

$$K = \frac{\omega_o \sigma}{2\pi} \tag{4.29}$$

The lower right corner of figure 4.6 shows the current values of ω_o and K, as well as the maximum disparity measurable with that frequency. The input images are shown on the upper left corner, and the resulting disparity in the lower left quarter. The upper right corner shows the confidence value $q(x)$ (cf. eq. 4.27), as well as the power $p(x)$ (cf. eq. 4.28).

Beyond the fact that the algorithm succeeds in recovering the two disparities in the image, figure 4.6 shows clearly the main characteristics of the algorithm, as discussed in page 75. The resulting disparity function $d(x)$ is continuous, smoothly varying from a disparity of 5 on the left to a disparity of 10 in the right, thus "filling" the disparity values in between the two peaks. The confidence value is lower in the center, where there is a transition from one "sinc" to the "other", as it is indeed the place where the information on the left and right images is more incoherent. Finally, the power signal has its maxima at the position of the "sinc" peaks, as it is the place where the biggest contrast in the original images can be found.

Figure 4.7 shows the result of the same Gábor function when applied to a noisy version of the same test images (with additive noise of 20 dB SNR.) The algorithm is robust enough to find the correct disparity.

4.2.4 Experiment II: A door

In the second experiment carried out using the stereo algorithm based on the Gábor phase (here referred to as algorithm I) a Khepera robot equipped with a `gemini` stereovision turret (cf. section 7.2) is placed in front of a wall with a small passage or door. The wall and the surface behind the door are marked with similar black stripes 2 cm wide, as seen in figure 4.8. The robot can use the disparity to find out the existence of a passage.

When applying the images to the Matlab implementation of the algorithm I, the optimal parameters are found to be $\omega_o = 0.328$ and $K = 0.5$. The result (see figure 4.9) shows that the door is detected as a "valley" in the center of the disparity image. The saturation of the input images has certainly had a beneficial effect, by eliminating the noise in the image. The disparity measurements on the borders of the image should be rejected altogether. They correspond to monocular views (i.e., they are only seen by one of the retinas) and therefore they produce false matches. Moreover, as it can be seen in the original images, the lens in the optic system attenuates the images on the borders, making the measurement even less accurate.

On-Khepera processing

The Gábor I algorithm has also been applied on the Khepera robot, that is, the computation is actually performed on the `ext331` turret, equipped with a MC68331 microprocessor. To do

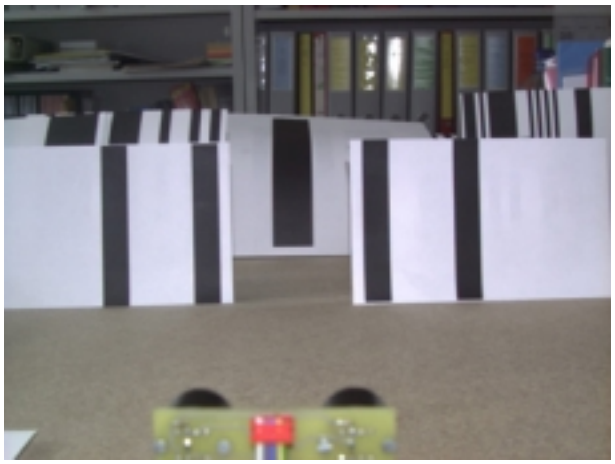


Figure 4.8: The Khepera robot, equipped with the `gemini` turret (cf. sec 7.2) tries to find out a passage in a wall covered with black stripes.

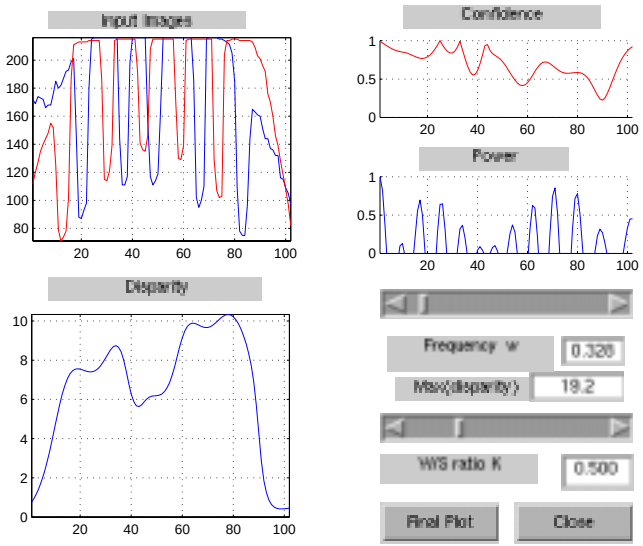


Figure 4.9: The stereo algorithm based on the Gábor phase finds the disparity (i.e., depth) in the scene seen in figure 4.8.

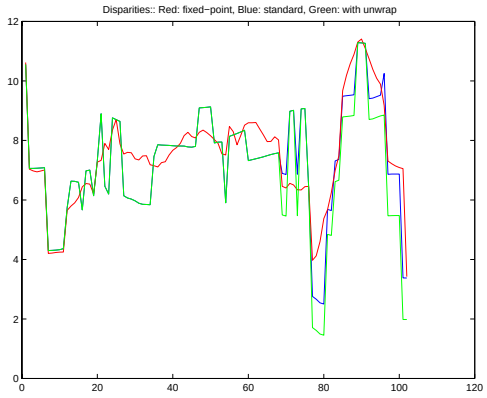


Figure 4.10: Comparison of the fixed-point version of Gábor I algorithm with the floating-point one. The floating-point one is the smoothest, but the fixed-point version approximate a discretization of the floating-point one. Two fixed-point versions have been tried, with unwrap() and without unwrap(), and the results coincide everywhere except at the right of the image.

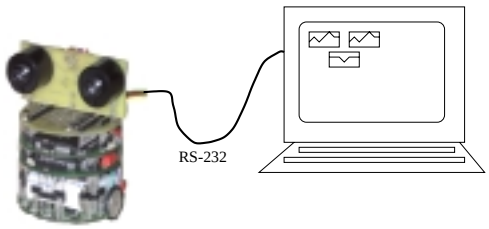


Figure 4.11: The Khepera robot, equipped with the `gemini` stereovision turret acquires two images and computes disparity on the `ext331` turret. The resulting disparity is sent to the workstation via serial cable for displaying purposes.

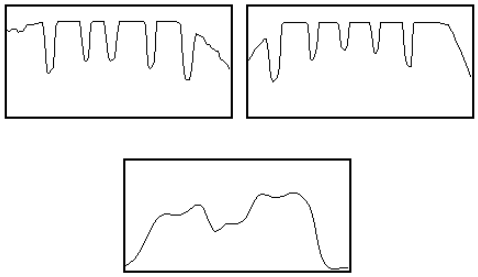


Figure 4.12: Screen shot showing the input images (top) and the calculated disparity as computed on the Khepera robot (bottom).

so, the algorithm described in page 78 has been coded in C, using fix-point notation to make it as fast as possible (see [Sav99] for the details.) To further reduce the computation time, the phase is not unwrapped. Figure 4.10 compares the results of the floating-point, fixed-point with `unwrap()` and fixed-point without `unwrap()` versions of the Gábor I algorithm for a test image. Besides small differences due to the phase wrapping, the fixed-point version compares favorably with the floating-point one.

The resulting disparity is sent through the serial cable to workstation, where it is displayed (see diagram 4.11).

Figure 4.12 shows the screen shot with the input images (upper two windows) and the resulting disparity (lower window). As it can be seen, it does not differ greatly from the Matlab (i.e., floating-point) version shown in figure 4.9.

4.2.5 Experiment III: a slanted surface

A third experiment has been conducted with the `gemini` turret facing a slanted surface with periodic grating, as shown on the left of figure 4.13. The grating consists of a superposition of several spatial frequencies, with a maximal contrast of 57%.

Using `gabordemo`, the optimal parameters are found to be $\omega_o = 0.42$ and $K = 0.92$. The right half of figure 4.13 shows the two input images that are obtained under those conditions. The two bottom-right images show the resulting disparity and depth, compared with the true disparity and depth (indicated with dotted lines).

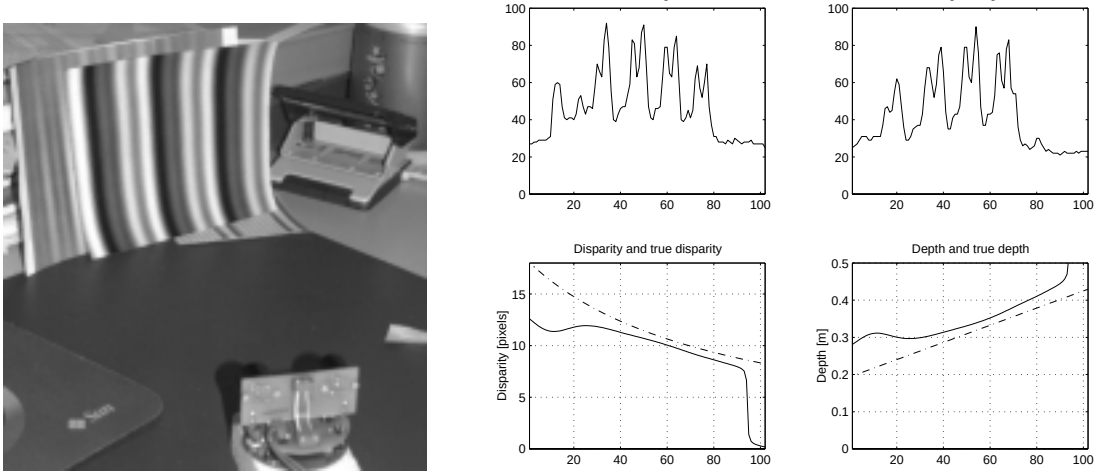


Figure 4.13: On the left, the `gemini` turret is facing a slanted grating at around 30 cm. On the right, the corresponding input images (top) and the calculated disparity and depth (bottom), together with the true (measured) depth and disparity, shown in dotted lines.

Depth has been computed from the resulting disparity by applying equation 3.13 with the focal length and base line corresponding to the `gemini` turret (see table 7.1). Disparity d is multiplied by the pixel pitch (85 μm) to render equation 3.13 homogeneous. True depth has been computed by measuring the setup in figure 4.13, with a 2 mm accuracy.

As it can be seen, the Gábor algorithm succeeds in recovering a disparity function that is fairly correct. There is a certain offset between the true disparity and the computed one, but then again, the true depth has been measured with limited accuracy. The important result shown in the image is that the Gábor algorithm is able to perceive the surface and that the actual angle of inclination of the surface is correctly recovered.

4.2.6 Influence of image contrast

To study the influence of contrast on the disparity recovery of the Gábor algorithm, the contrast in the input images has been modified by scaling and the Gábor algorithm has been applied on the resulting images with the same frequency and spatial spread σ.

Figure 4.14 shows the result of the contrast variation on the door image (cf. fig. 4.8 and 4.9). The left half shows the disparity maps for different contrast levels, measured as the ratio

$$c = \frac{I_{max} - I_{min}}{I_{max} + I_{min}}$$

The original image, as shown in figure 4.8, is drawn in solid line, and it has a 48% contrast.

The relative error, shown on the right half of figure 4.14 is measured as the difference between the disparity and the disparity obtained for the original, 48%-contrast image. The errors on the right border of the image have not been taken into account, because it is an area without overlap between the left and right images.

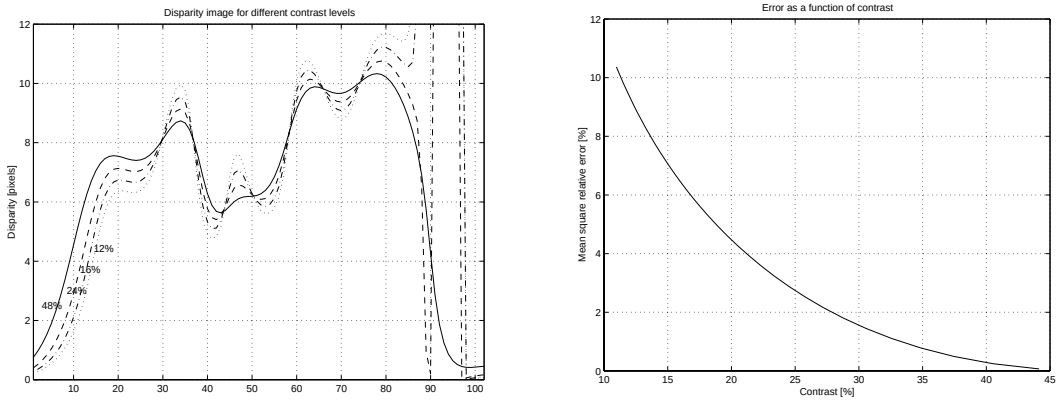


Figure 4.14: On the left, variations in disparity recovery as a function of the input image contrast, for the door image (cf. 4.9). On the right, the mean-square relative error is plotted against the image contrast.

Figure 4.15 shows the result of the same experiment on the slanted-surface image (cf. fig. 4.13). the left half shows the disparity computation for each contrast level, and the right

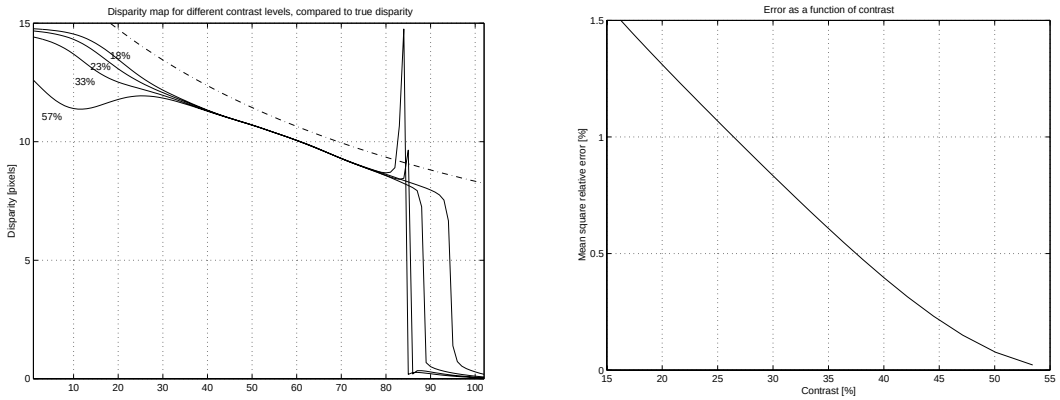


Figure 4.15: On the left, variations in disparity recovery as a function of the contrast levels in the image. Results on the image of the slanted surface (fig. 4.13). On the right, the mean-square relative error is plotted against the image contrast.

half shows the mean-square relative error, as compared to the true disparity.

The disparity estimation in the central region, which corresponds to the area of overlap between left and right images, does not change with contrast. Variations are concentrated on the left and right borders, where confidence is lower.

These results show that the Gábor algorithm is robust with respect to changes in contrast. Although low-contrast images introduce an error in the estimated depth, this error is small, and it does not affect the a posteriori interpretation of the image. For low contrast levels, a door can still be seen in figure 4.14 and a slanted surface can be distinguished in figure 4.15.

4.2.7 Effects of occlusions

Equation 4.19, which supposes that the left and right images contain the same information, has been used as a starting hypothesis in the Gábor algorithm. This hypothesis is not true in the presence of *occlusions* and heavy noise. It is therefore convenient to test the robustness of the algorithm when facing occlusions.

To that end, the images obtained by placing the `gemini` turret in front of a slanted grating (fig. 4.13) have been taken, and an artificial occlusion has been introduced. As it can be seen in figure 4.16 one of the pulses on the left image has been completely eliminated and replaced by a constant value. In principle, this would affect the matching with the corresponding projection on the right image.

In the lower half of figure 4.16 the recovered disparity and corresponding depth are shown and compared with the original disparity and depth (indicated with dotted lines.) The correct disparity is recovered, inasmuch as the presence of a slanted surface at a distance of around 30 cm is detected. In other words, the resulting disparity map would provide the same interpretation of the scene.

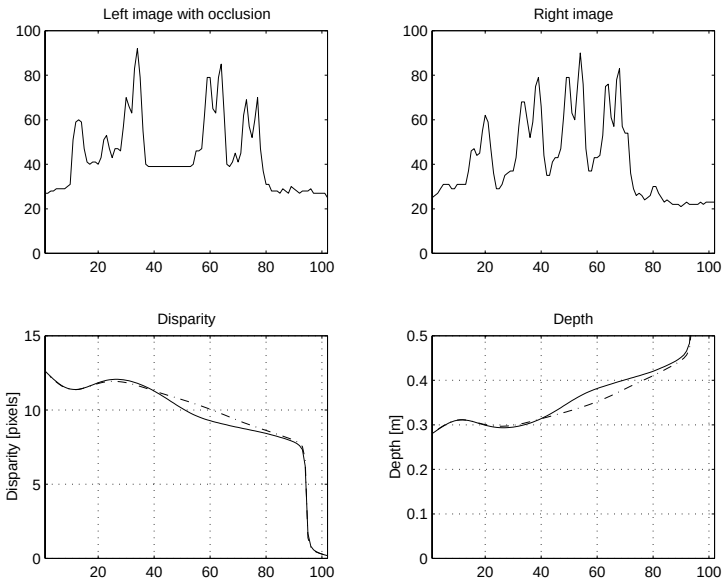


Figure 4.16: Effects of occlusions on the disparity recovery.

As already advanced in page 75 item 2, the algorithm provides a continuous disparity function. This implies that, in case of occlusions, disparity is estimated by interpolation.

4.3 Algorithm II

The previous experiments have shown that the Gábor phase is indeed capable of recovering the disparity in a pair of stereo images. It suffices to tune the algorithm to the right parameters ω_o and σ (or K). However, there is not a single combination (ω_o, σ) that will fit all possible images for a given stereoscopic system (e.g., the `gemini` turret). Firstly because the modulating frequency ω_o is related to the maximum disparity, and hence it will be somehow dependent on the contents of the scene. Secondly because σ is related to the amount of smoothing, that is, the amount of noise and disparity interpolation. Therefore, it can be expected that different regions in the scene will be better processed with different combinations (ω_o, σ).

We can not generate a Gábor function which is made of different combinations of (ω_o, σ), but we can certainly try to combine the outputs of the algorithm for different combinations of frequency and variance. To do so we naturally take advantage of the confidence value $q(x)$ (see eq. 4.27). One of the possibilities, which has been tried in [Sav99] is that of taking, for each pixel, the disparity of the (ω_o, σ) combination that yields the highest confidence value at that pixel. If there are i Gábor filters that are applied, the resulting disparity is taken as

$$d(x) = \{d_i(x) \,|\, q_i(x) > q_j(x) \forall i \neq j\} \tag{4.30}$$

A weighted average of disparities has been used, as in [San88]. In this case, each disparity is

weighted by its confidence factor, so that disparities with low confidence have a smaller influence on the final result. The actual formula for the calculation of the final disparity is

$$\overline{d(x)} = \frac{\sum_{i=0}^{N} d_i(x) q_i(x)}{\sum_{i=0}^{N} q_i(x)} \tag{4.31}$$

where $d_i(x)$ is the disparity computation (according to eq. 4.25) for the i^{th} combination, $q_i(x)$ is its corresponding confidence level (cf. eq. 4.27) and N is the number of (ω_o, σ) combinations that are applied on the stereo pair of images.

If most $d_i(x)$ seem to be in agreement, it can be inferred that the resulting disparity $\overline{d(x)}$ is most probably right, or correspondingly, that there is a high confidence in the resulting value. The standard deviation being a good measure of the agreement among different values, a combined confidence level can be defined as

$$s(x) = \frac{1}{1 + \frac{\sum_{i=0}^{N} (d_i(x) - \overline{d(x)})^2 q_i(x)}{\sum_{i=0}^{N} q_i(x)}} \in [0, 1] \tag{4.32}$$

Therefore, the algorithm (referred to as Gábor II) for the combined disparity can be written as in algorithm 4.2, page 88.

4.3.1 Experiment IV: Multifrequency Gábor

The multiple-frequency Gábor algorithm, referred to as Gábor II algorithm, has been tested on the same image as experiment II (cf. page 81). A set of 5 frequencies around the optimum frequency used in experiment II has been defined, and the parameter K is kept constant, (σ varies according to eq. 4.29) so that each Gábor function sweeps a constant number of octaves.

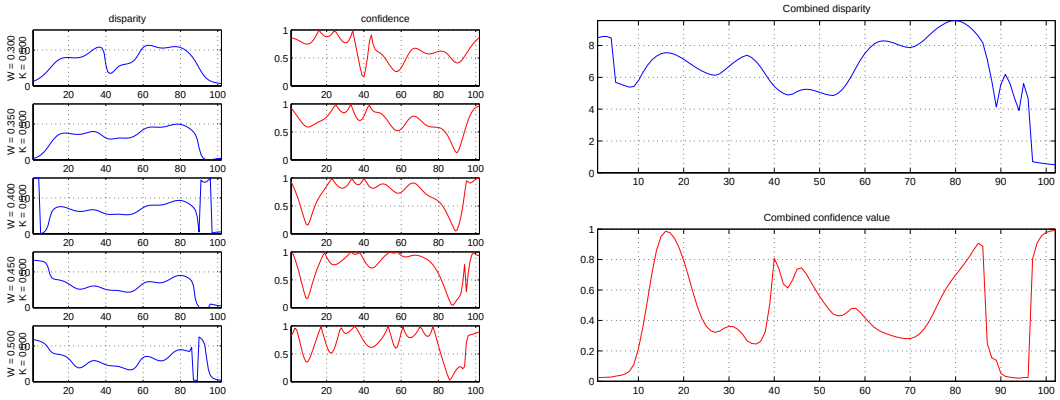


Figure 4.17: Gábor II: On the left, the 5 individual disparities and corresponding confidence values. On the right, the combined disparity function and confidence value.

For this experiment the values that have been chosen are

ω_o	0.3	0.35	0.4	0.45	0.5
K	0.5	0.5	0.5	0.5	0.5

Algorithm 4.2 Algorithm for the recovery of disparity d in the $left$ and $right$ images using several Gábor functions (Algorithm Gábor II)

```
[d] = gabormulti(left,right)
  {
    for i=0:N,
      g  = gabor(w(i),sigma(i));
      cl = convolution(left,g);
      cr = convolution(right,g);
      eps = 1e-6;
      qa = (eps+abs(cl))./(eps+abs(cr));
      qb = (eps+abs(cr))./(eps+abs(cl));
      q(i) = min(qa,qb);
      fl = unwrap(angle(cl));
      fr = unwrap(angle(cr));
      dx = fr - fl;
      dx = dx + 2*pi*(dx<0);
      d(i) = dx/omega;
    end;
    for i=0:N,
      d = d + d(i)*q(i);
      a = a + q(i);
    end;
    d = d/a;
    for i=0:N,
      s = s + (d(i)-d)^2*q(i);
    end;
    s = 1/(1+s/a);
    return d;
  }
```

Figure 4.17 shows the resulting disparity image when several Gábor filters are used as described in page 88. The resulting disparity is not better than the optimum disparity found in figure 4.9, as it is simply an average of sub-optimal disparity solutions. Still the resulting disparity is mostly correct and the Gábor algorithm eliminates the need for ad-hoc tuning. It suffices to sweep the space of (ω_o, K) to obtain a satisfactory result. Of course, computation time increases linearly with the number of (ω_o, K) combinations that are considered.

4.4 Gaussian expansion

The Gábor function has simple mathematical expression, but still it is a complicated function to generate in some simple systems, particularly in analog VLSI circuits and other physical realizations, because of the negative lobes. Therefore it would be highly desirable to recreate the Gábor function in a simpler way, or at least to reproduce the results of the Gábor I and II algorithms. A very convenient equivalent function would be an expansion into Gaussian

elementary signals. Such an approach has been proposed in [GAB46] and proved in [BAS80] for the exact reproduction of the original signal. Unfortunately, the Gaussian elementary signals are not orthonormal and therefore the determination of the expansion coefficients is difficult.

In the frame of this project however, a high-fidelity expansion is not needed, and a simple approximation would be sufficient because

1. Only a particular function needs to be approximated. A general method for the approximation of any function is not needed, as we are only concerned with the Gábor function.

2. The approximation does not require the utmost precision because it is not intended for the reconstruction of the original signal. In our case we are mostly interested in recovering the phase at the output, and therefore small changes in magnitude are not too important.

3. The approximation is part of a noisy processing system. The error introduced by the approximation may be smaller than the noise in the acquired images. Furthermore, if it were implemented in an analog system whose output is to be digitized at some point[4], it must be borne in mind that precision in this systems is limited to 8 to 10 bits (cf. section 2.2.2).

4. Out of the whole family of Gábor functions, the algorithm described above (sec. 4.2) seems to work more efficiently with simple Gábor functions, that is, functions with a small number of lobes. For instance, the optimum Gábor function found in experiment II (cf. page 80), shown in figure 4.18a, has only 5 lobes.

With these premises, the Gábor function $g(x)$ can be approximated by a limited number of Gaussian functions according to :

$$g_{\omega_o,\sigma}(x) \approx \widetilde{g}\ (x) = \sum_{n=-N}^{N} (-1)^n a_n \cdot gau\beta(x - c_n, \sigma_s) + j \sum_{m=-M}^{M} (-1)^m b_m \cdot gau\beta(x - d_m, \sigma_s) \quad (4.33)$$

where a_n, b_m, c_n, d_m and σ_s are real coefficients.

Each lobe of the Gábor function is approximated by a Gaussian function. The width of the Gábor lobes depends on the modulating frequency, and it is therefore equal to the distance between two zero-crossings, that is, $x = \pi/\omega_o$. The width of a Gaussian function is in turn approximated by $x = 4\sigma_s$, because $erf(2) = 0.995$[5] . Thus, all the elementary Gaussian functions are found to have the same standard deviation σ_s.

The coefficients a_n and b_m are then easy to determine, as they are the value of the Gaussian envelope of the Gábor function, evaluated at the peaks of the modulating sine and cosine, which happen at $x = n\pi$ for the real part and $x = (2m+1)\frac{\pi}{2}$ for the imaginary part. Similarly, the coefficients c_n and d_m correspond to the same maxima of the sine and cosine modulating functions.

[4] In almost all small mobile robots motors are controlled by a digital microprocessor, so an A/D conversion is to be expected at some stage.

[5] The error function is defined as $erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt$

The last parameters to determine are N and M, the number of Gaussian elementary functions. As in the lines above, the Gaussian envelope of the Gábor function can be considered to vanish at a distance of two times the standard deviation from the maximum, and therefore the last term is for $c_N \approx d_M = 4\sigma$. To summarize, the parameters are calculated as follows:

$$\sigma_s = \frac{\pi}{4\omega_o} \tag{4.34}$$

$$N = \frac{4\sigma\omega_o}{\pi} = 8K \tag{4.35}$$

$$M = \frac{1}{2}(\frac{8\sigma\omega_o}{\pi} - 1) \approx 8K \tag{4.36}$$

$$a_n = e^{-\frac{n^2\pi^2}{2\sigma^2\omega_o^2}} = e^{-\frac{n^2}{8K^2}} \tag{4.37}$$

$$c_n = \frac{n\pi}{\omega_o} \tag{4.38}$$

$$b_m = e^{-\frac{(2m+1)^2\pi^2}{8\omega_o^2\sigma^2}} = e^{-\frac{(2m+1)^2}{32K^2}} \tag{4.39}$$

$$d_m = (2m+1)\frac{\pi}{2\omega_o} \tag{4.40}$$

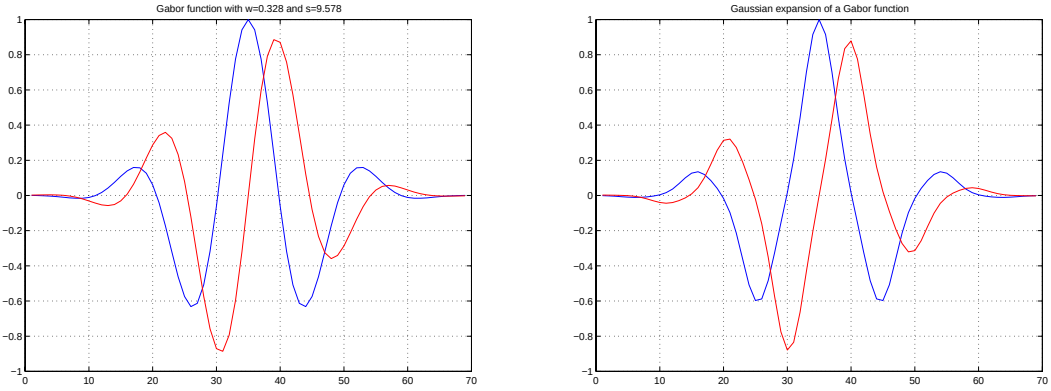


Figure 4.18: (A) The original Gábor function (real and imaginary parts). (B) The same function approximated by Gaussian expansion.

The benefits of such a simple expansion can be clearly seen when applied to the Gábor phase stereo algorithm. If the expansion in equation 4.33 is written in a more general way as

$$\widetilde{g}\ (x) = \sum_{k=-K}^{K} a_k \cdot gau\beta(x - b_k, \sigma_s) \tag{4.41}$$

with a_k being a complex coefficient, and this approximation is applied to equations 4.20 and 4.21 we obtain

$$\widetilde{c}\ (x) = s(x) * \widetilde{g}\ (x) = \sum_{k=-K}^{K} a_k\, s(x) * gau\beta(x - b_k, \sigma_s) = \sum_{k=-K}^{K} a_k\, m(x - b_k) \tag{4.42}$$

Thus, the desired output $\widetilde{c}\ (x) \approx c(x)$ is a series of shifted and scaled functions $m(x)$, where

$$m(x) = s(x) * gau\beta(x - b_k, \sigma_s) \tag{4.43}$$

Notice that equation 4.43 is the smoothing of the input signal $s(x)$ (i.e., the image) with a Gaussian kernel, an operation that is easier to implement on physical systems. Moreover, if $s(x)$ is positive and real, so is $m(x)$.

4.4.1 Experiment V

To compare the results of the Gaussian approximation with the original Gábor function, an experiment has been carried out on the same stereo pair as the previous experiments. The steps described in the Gábor I algorithm (cf. page 78) are followed, except that instead of using $c(x)$, the images are convolved with a Gaussian basic function and the resulting $m_l(x)$ and $m_r(x)$ functions are shifted and added according to eq. 4.42 to produce $\widetilde{c}\ (x)$. Because of the discrete nature of the spatial coordinates of the image, all shifts b_k have been rounded to the nearest integer. A more precise reconstruction could be obtained if, by interpolating the m_l and m_r intermediate images, subpixel shifts were possible. The introduction of an interpolation stage would increase the computation time, while results have shown that the discretization of b_k does not reduce noticeably the performance of the algorithm. In the last step, as in the previous algorithms, the phase of $\widetilde{c}\ (x)$ is recovered to measure the disparity.

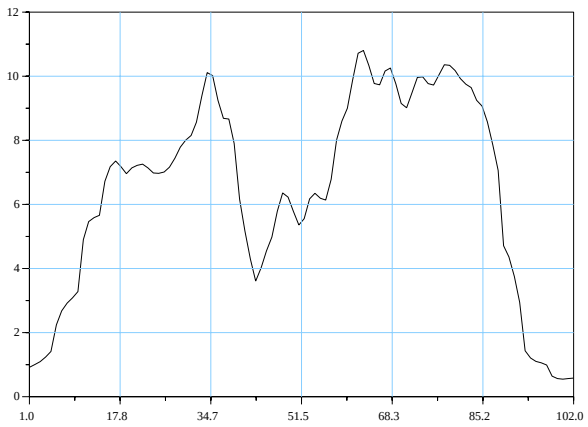


Figure 4.19: Result of the application of the Gaussian expansion shown in figure 4.18b to the disparity calculation in image 4.8. To be compared with the result of the actual Gábor function shown in figure 4.9.

Figure 4.19 shows the result of applying the Gaussian expansion of the Gábor function seen in figure 4.18b to the images taken with the Khepera robot facing a passage in a wall, as shown in figure 4.8. Although noisier than the disparity calculation obtained with a Gábor function, the approximation is effective enough to show a region of lower disparity that betrays a passage in the wall.

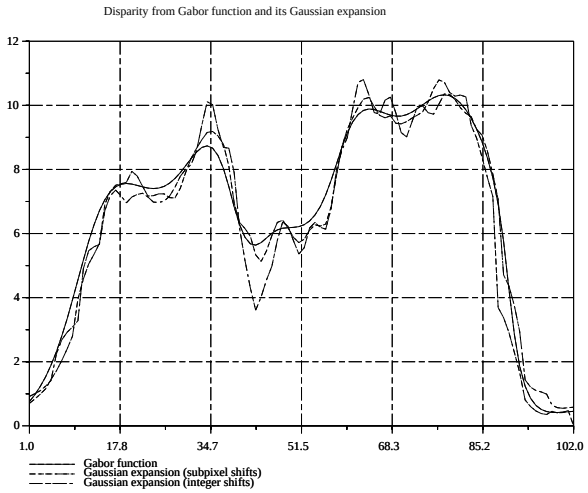


Figure 4.20: Comparison of disparity results for the original Gábor function with that of a Gaussian expansion with subpixel shifts and with integer shifts.

4.5 Exponential expansion

For a realization of these algorithms in analog VLSI the first obstacle to be solved is the implementation of a filter with a Gaussian impulse-response. A Gaussian filter can be implemented as a resistive grid where each node is connected to its immediate neighbors with a weight (i.e., a conductance) of 1 and to second-nearest-neighbors with a weight of -4. In [KOB91] such a negative conductance is implemented with a *negative impedance converter* (NIC), a circuit of limited stability range that increases the area per pixel. However, as we have seen in section 2.4.4, diffusion networks can easily produce a bilateral, exponentially-decaying function, and such a function resembles the Gaussian function.

The exponential function $f(x,\lambda) = e^{-\frac{|x|}{\lambda}}$ that best approximates the Gaussian $g(x,\sigma_s)$ in the least square error sense is the exponential with

$$\lambda = 1.312\,\sigma_s \tag{4.44}$$

A proof for equation 4.44 can be found in appendix 8.2 at page 149.

The expansion of the Gábor function into Gaussians (cf. eq. 4.41) can be written as series expansion of bilateral exponentials

$$\widetilde{g}\,(x) = \sum_{k=-K}^{K} a_k \cdot e^{-\frac{|x-b_k|}{1.312\sigma_s}} \tag{4.45}$$

and the same procedure described in equation 4.42 applied to the new function

$$m(x) = s(x) * e^{-\frac{|x|}{1.312\sigma_s}} \tag{4.46}$$

Such a function $m(x)$ can be seen as the result of the application of a diffusion filter on the input image $s(x)$.

4.5.1 Experiment VI

To test the effectiveness of the bilateral exponential expansion a test was conducted on the same input image as experiment IV. The coefficients of the expansion (a_k,b_k) were calculated as for the Gaussian expansion that is, according to equations 4.35 through 4.40. The input images were convolved with an exponentially-decaying kernel of $\lambda_s = 1.312\sigma_s$, and the $\widetilde{c_l}\ (x)$ and $\widetilde{c_r}\ (x)$ function were obtained by shifting and scaling the convolutions $m_l(x)$ and $m_r(x)$ according to equation 4.42. Eventually, the Gábor phase was recovered as in the previous cases.

The resulting disparity calculation is plotted in figure 4.21 together with the reference disparity (i.e., calculated with the actual Gábor function). The disparity result using the exponential expansion (shown in solid line) compares favorably with the disparities obtained with the Gaussian expansion (dashed line) and the actual Gábor function (dash-dotted line).

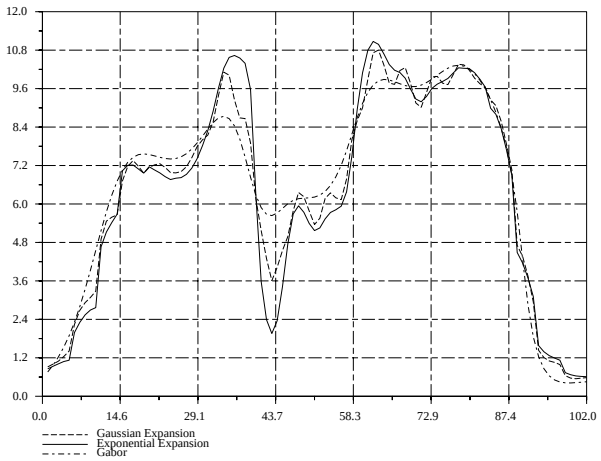


Figure 4.21: Disparity result using an exponential expansion to approximate the Gábor function. The result is comparable to the Gaussian expansion and the actual Gábor function.

4.6 Pseudogabor function

Gábor functions, either in the original formulation (eq. 4.8) or as a Gaussian expansion (eq. 4.33) have the inconvenient of being complex-valued functions. At best this means that all convolutions must be performed twice: one for the real part and one for the imaginary one. In a hardware implementation, such an approach requires twice as much area in analog circuits or twice the time and memory in digital circuits [Blo97].

It would be very convenient to find a way of operating exclusively with real functions. Such a function could be the real part of a Gábor function of arbitrary phase, called Gábor cosine function [Blo97]

$$h(x) = e^{-\frac{x^2}{2\sigma}} \cos(\omega_o x + \varphi) \tag{4.47}$$

or better yet for the purpose of a hardware implementation, its exponential approximation [Raf98], the pseudogabor function

$$f(x) = e^{-\frac{|x|}{\lambda}} \cos(\omega_o x + \varphi) \tag{4.48}$$

with $\lambda = 1.312\,\sigma$ (see appendix 8.2, page 149, for a proof of this approximation). Since both functions are fundamentally equivalent, the developments that follow will use the exponential approximation. Thus all results will be true for $f(x)$ to a certain approximation error, and correct for $h(x)$.

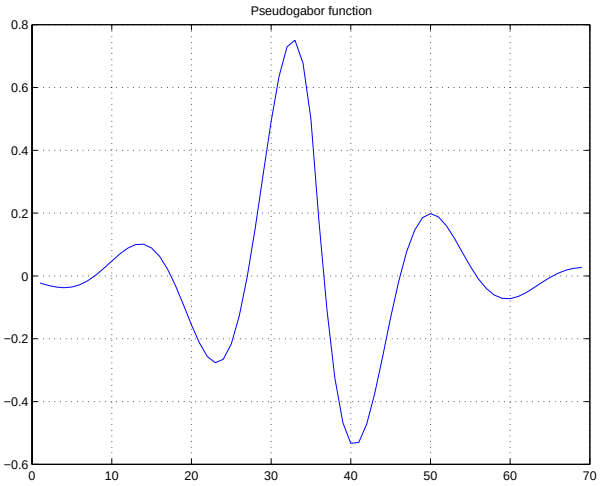


Figure 4.22: A pseudogabor function of arbitrary phase.

Relationship with the Gábor function

The pseudogabor function has no phase (or it has zero phase, since it is real), and therefore the algorithms discussed in section 4.2 is not applicable anymore. It is therefore necessary to adapt or modify the algorithm for use with pseudogabor functions. To do so, let us first discuss the relationship between pseudogabor and Gábor functions, and next introduce the new stereo algorithm.

The relationship with the Gábor function is straightforward, as equation 4.48 can be developed as

$$f(x) = e^{-\frac{|x|}{\lambda}} \cos(\omega_o x + \varphi) = e^{-\frac{|x|}{\lambda}} \cos(\varphi)\cos(\omega_o x) - e^{-\frac{|x|}{\lambda}} \sin(\varphi)\sin(\omega_o x) \tag{4.49}$$

$$\approx \cos(\varphi)\mathrm{Re}\{g(x)\} - \sin(\varphi)\mathrm{Im}\{g(x)\} = \cos(\varphi) g_R(x) - \sin(\varphi) g_I(x) \tag{4.50}$$

where the Gábor function $g(x)$ (cf. eq. 4.8) is written as

$$g(x) = g_R(x) + j\, g_I(x) = e^{-\frac{x^2}{2\sigma}} \cos(\omega_o x) + j\, e^{-\frac{x^2}{2\sigma}} \sin(\omega_o x) \tag{4.51}$$

With these equivalences in hand, we can set forth to analyze the Gábor algorithm as set in equations 4.20 and 4.21. For the left image we compute

$$c_l(x) = g(x) * l(x) = (g_R(x) * l(x)) + j\,(g_I(x) * l(x)) \tag{4.52}$$

whereas for the right image we have

$$c_r(x) = g(x) * r(x) = |c_r| e^{j\varphi_r(x)} \approx |c_l| e^{j(\varphi_l(x)+\omega_o d)} = |c_l| e^{j\varphi_l(x)} e^{j\omega_o d} \tag{4.53}$$

as it was already shown in equation 4.24. This in turn can be developed into

$$\begin{aligned} c_r(x) &= c_l(x)\{\cos(\omega_o d) + j\, \sin(\omega_o d)\} \\ &= \{(g_R(x) * l(x)) + j\,(g_I(x) * l(x))\}\{\cos(\omega_o d) + j\, \sin(\omega_o d)\} \\ &= (g_R(x) * l(x)) \cos(\omega_o d) - (g_I(x) * l(x)) \sin(\omega_o d) \\ &\quad + j\,\{(g_I(x) * l(x)) \cos(\omega_o d) + (g_R(x) * l(x)) \sin(\omega_o d) \qquad (4.54) \\ &\equiv (g_R(x) * r(x)) + j\,(g_I(x) * r(x)) \qquad (4.55) \end{aligned}$$

By taking the real part of the equivalence between equations 4.54 and 4.55 we obtain

$$(g_R(x) * r(x)) = (g_R(x) * l(x)) \cos(\omega_o d) - (g_I(x) * l(x)) \sin(\omega_o d) \tag{4.56}$$

and applying equation 4.50 into 4.56 we obtain

$$(f(x) * r(x))_{|\varphi=0} = (f(x) * l(x))_{|\varphi=\omega_o d} \tag{4.57}$$

Equation 4.57 indicates that for each pixel x in the image, the convolution of a pseudogabor function $f(x)$ with the right image matches with the convolution of the left image with a pseudogabor function whose phase is $\omega_o d(x)$. In other words, the disparity $d(x)$ at pixel x is such that it minimizes the error function

$$E = [(f(x) * r(x))_{|\varphi=0} - (f(x) * l(x))_{|\varphi=\omega_o d(x)}]^2 \tag{4.58}$$

Experiment VII

This experimental set was carried out with the interactive Matlab function `gabordemo3`, which lets the user modify the three free parameters in equation 4.58: ω_o, λ and φ_l. Figures 4.23 through 4.26 show the screen shot of two such sessions. The upper left corner of the image displays the input images. The lower left corner shows the result of the convolutions $c_l(x) = f(x, \varphi) * l(x)$ and $c_r(x) = f(x, 0) * r(x)$. On the upper right the normalized error and the power are shown. Normalized error is defined as

$$\overline{e}(x) = \frac{|c_l(x)|^2 - |c_r(x)|^2}{|c_l(x)|^2 + |c_r(x)|^2} \tag{4.59}$$

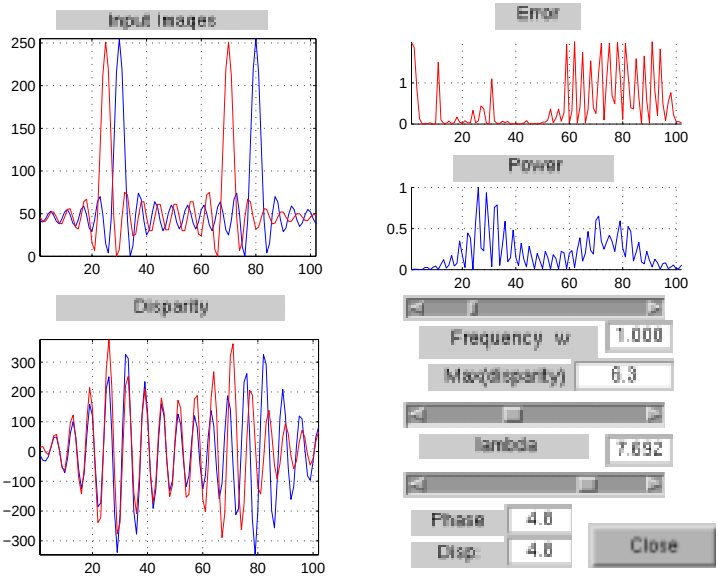


Figure 4.23: Disparity computation with pseudogabor functions: When setting the phase to 4.8 rad, a perfect match is found for the left peaks. Disparity there is estimated to be 4.8 pixels. Notice how the error in the left half of the image is low. Continues on figure 4.24

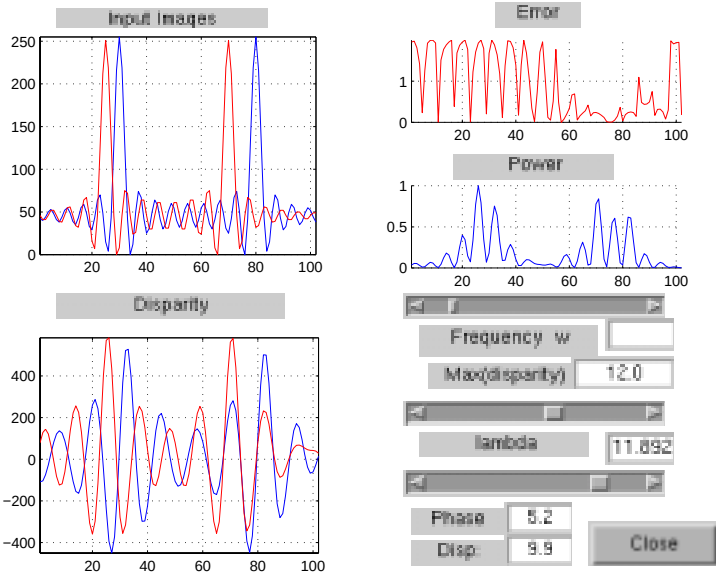


Figure 4.24: When setting the phase to 5.2 rad, a perfect match is found for the right peaks. Disparity there is estimated to be 9.9 pixels and warranted by a low error on the right half of the image.

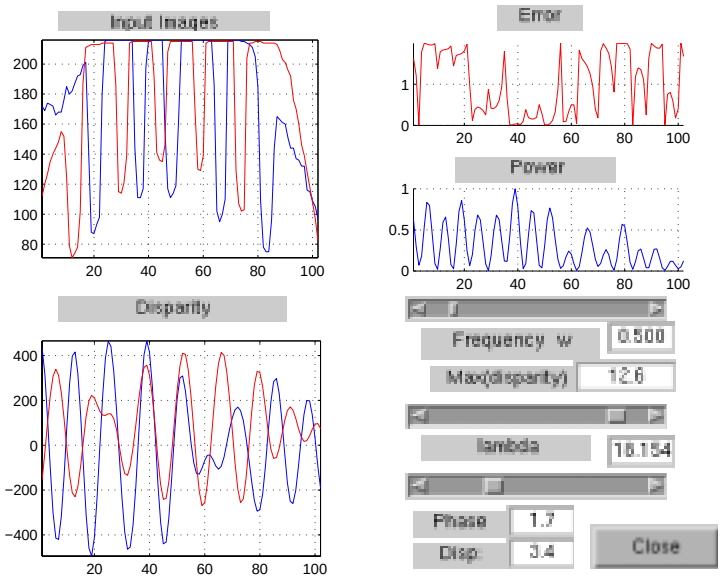


Figure 4.25: Disparity calculation on a real image: By setting the phase to 1.7 rad, a good match is found in the central region of the image (pixels 40 to 50), where the door is expected to be. Disparity is estimated at 3.4 pixels.

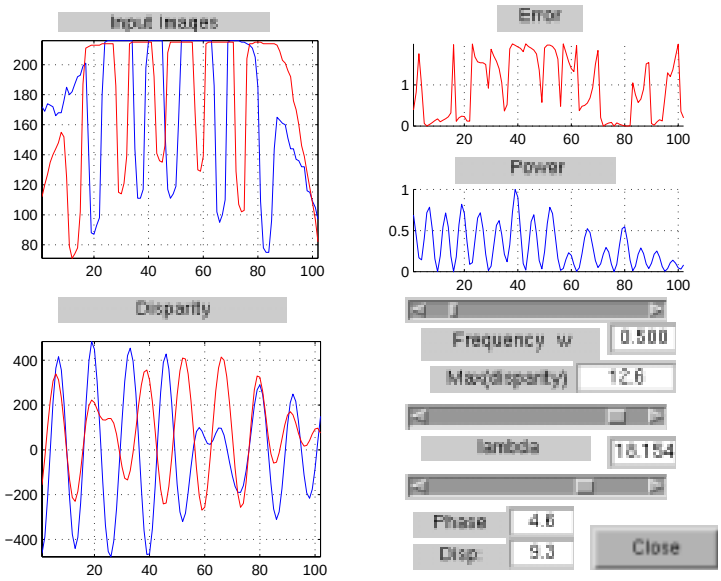


Figure 4.26: A phase of 4.6 rad produces a good match (low error) on the outer regions of the image (pixels 70 to 80 and around 20). Disparity there is estimated at 9.3 pixels.

The lower right corner shows the current values for ω_o, λ and φ. It also displays the disparity estimated from the current frequency and phase.

The first test is conducted on a test image consisting of two peaks at disparities 5 (left peak) and 10 (right peak). Figures 4.23 and 4.24 show that the two disparities were recovered.

The second test was performed on a real image taken by the Khepera robot. The robot is facing a passage in a wall (cf. figure 4.8), and therefore a region of smaller disparity (i.e., greater depth) is to be expected in the center of the image. Figures 4.25 and 4.26 show the recovery of the disparity in the passage ($d \approx 5$ pixels) and around it ($d \approx 10$ pixels).

4.7 On the circuit implementation of the Gábor algorithms

An analog-circuit implementation of the Gábor phase algorithms requires solving the problem of convolving the image —in this case with a Gábor function— and recovering the phase.

As discussed in section 2.4.4, resistive networks can be used to filter the images with convolution masks. Filters with Gábor-like impulse responses can be obtained with these techniques too. Raffo et al. [RAF96] [RAF98] have shown that a resistive network with second-neighbor connections can be combined to generate a Gábor impulse-response filter.

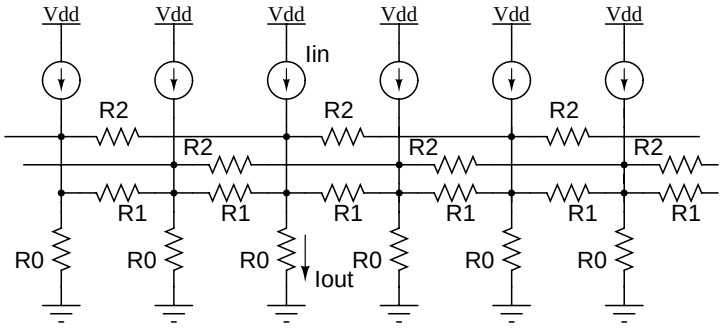


Figure 4.27: A second-order resistive networks, which can be used to obtain a Gábor impulse-response.

A second-order resistive network as in figure 4.27, where the second-neighbor connections are inhibitory (i.e., $R_2 < 0$), has an impulse response that resembles the real part of a Gábor function. Each node in the network obeys to equation 4.60

$$I_{out}(x) = I_{in}(x) + G\, I_{out}(x-1) + G\, I_{out}(x+1) - K\, I_{out}(x-2) - K\, I_{out}(x+2) \qquad (4.60)$$

where $I_{in}(x)$ is the input image at the x^{th} pixel, I_{out} is the output current (measured on the R_0 resistors) and G and K are gains that depend on R_1/R_0 and R_2/R_0. The result I_{out} is a convolution of the input image I_{in} with a kernel of the form [RAF98]

$$h(x) = Ce^{-\frac{|x|}{\lambda}}[(1+e^{-\frac{2}{\lambda}})sin(\omega_o)cos(\omega_o x) + (1-e^{-\frac{2}{\lambda}})cos(\omega_o)sin(\omega_o|x|)\,] \qquad (4.61)$$

The analog VLSI circuit proposed by Raffo et al. to implement equation 4.60 does not use resistors, but rather current mirrors with fixed gains (cf. 2.4.2). Since currents on a node can be positive and negative, the circuit is split into two symmetric halves that process positive and negative currents separately. Figure 4.28 shows the positive-half circuit in such a node [Bis97].

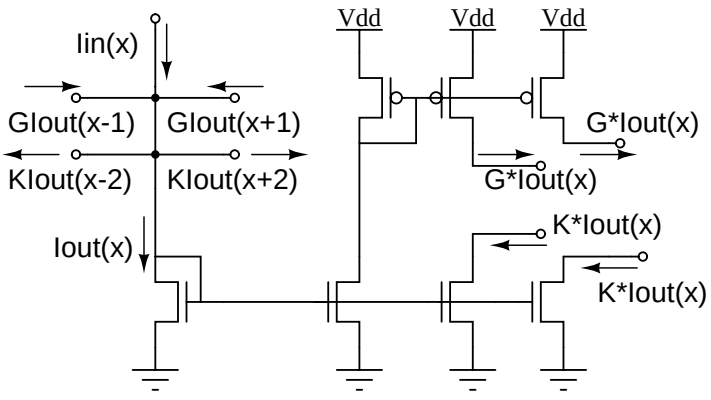


Figure 4.28: A positive-current half of a node in the Gabor network. From [Bis97]

4.7.1 Gábor function

To generate the real and imaginary parts of a Gábor function $g(x)$, the outputs of the network must be combined according to

$$Re\{g(x)\} = -b_1\, h(x-1) + b_0\, h(x) - b_1\, h(x+1) \quad (4.62)$$

$$Im\{g(x)\} = b_1\, h(x-1) - b_1\, h(x+1) \quad (4.63)$$

where parameters b_0 and b_1 control the frequency and spread of the Gábor function [Raf98].

A circuit based on such Gábor networks can be used to convolve the input currents (i.e., the images) with a Gábor kernel. A single such circuit capable of generating the $h(x)$ kernel (cf. eq. 4.61) is necessary for each image and Gábor function. By combining the outputs of this network, the resulting coefficients $c_l(x)$ and $c_r(x)$ can be obtained.

The next step to be solved is the computation of the phase at each pixel. A phase-recovery computation, directly based on the arctan( ) function is not feasible in analog VLSI, because of the intrinsic periodicity and asymptotic behavior of the function.

A solution would be for a digital microprocessor to read the convolution output and perform the phase-recovery operation in the digital domain.

4.7.2 Pseudogabor function

Equation 4.61 describes a function very similar to the pseudogabor function defined in eq 4.48. By combining three such functions, a pseudogabor function of arbitrary phase can be produced by the appropriate choice of parameters b_{-1}, b_0 and b_1 according to equation 4.64 [Raf98].

$$f(x,\varphi) = b_1\, h(x-1) + b_0\, h(x) + b_{-1}\, h(x+1) \tag{4.64}$$

The circuit previously discussed can therefore be applied to the stereo algorithm based on the pseudogabor function (cf. section 4.6). A population of pseudogabor circuits of different phases can be used to test all of them against the zero-phase circuit, finally choosing the phase that minimizes equation 4.58. Such approach of course increases the circuit size, since a whole population of functions needs to be considered.

Alternatively, an adaptive matching mechanism based on a digital processor could be devised to match the phase, as in the circuit proposed by Raffo et al. [Raf95] [Bis96] to solve a similar stereo algorithm.

4.8 Summary

In this chapter the application of Gábor functions to the computation of disparity has been discussed. The relationship between Gábor functions and the Fourier transform have been underscored to justify if not prove their use in measuring the spectral phase-shift. Convolving an image with a Gábor function yields an approximation of the Fourier transform of the image evaluated at the Gábor function modulating frequency, and this property is exploited to measure the disparity between two images as a phase shift in the frequency domain.

A simple procedure for this calculation has been outlined and tested on a workstation and on a small mobile robot. The algorithm has been shown to efficiently recover the disparity between images and hence the depth in the scene.

The tests have confirmed the advantages of the Gábor phase algorithm as outlined in the theoretical discussion (cf. section 4.2). The algorithm provides a continuous disparity for all pixels, effectively interpolating at pixels of ambiguous disparity and in case of occlusions. It also avoids an explicit feature-extraction stage, thereby avoiding the difficult task of defining what is a feature, although both theory and tests have shown that there is an underlying preference for edges. Results have shown the robustness of this algorithm in front of occlusions and low contrast levels.

The use of local information is among the key advantages of Gábor functions and the main one when it comes to consider a VLSI implementation of such algorithm. Disparity is calculated based on the phase of each pixel and its correspondent one on the other image, and the pixel phase is dependent on the pixel input and that of its closer neighbors (neighborhood being defined by the Gábor function spatial spread. Therefore there is a reduced number of interpixel connections and this facilitates its implementation on silicon substrate, where connectivity is costly.

Limitations of the Gábor algorithm have also been studied and remedial solutions proposed. The algorithm produces a disparity measure for each pixel, although the accuracy of the measure is not uniform throughout the image. Hence a confidence value $q(x)$ has been defined, and the use of a power threshold recommended. Similarly, the algorithm is very sensitive to the choice of the modulating frequency and the spatial variance of the Gábor function, and a procedure

(cf. section 4.3) has been described to reduce the sensitivity by combining the results of several Gábor functions.

To simplify the implementation of the Gábor algorithms in digital and analog systems, the Gaussian and exponential expansions of the Gábor function have been proposed and tested, obtaining satisfying results. Similarly, a modified algorithm based on pseudogabor functions has been presented too, although its analog implementation remains to be developed.

Chapter 5

A differential stereoscopic algorithm

Contents

5.1 Introduction

This chapter will introduce a simple algorithm that can recover disparity using simple, feed-forward calculations. The simplicity of the algorithm affords a straightforward implementation as an analog VLSI circuit.

5.2 Algorithm

The differential algorithm is an intensity-based algorithm that matches regions on two images by using purely local information, stemming from the image intensities and their derivatives.

The algorithm was introduced by Lucas and Kanade [Luc81] as an "iterative image registration technique" for the matching of image regions of constant disparity. I have preferred to give it the name of "differential stereo", a variation on the name proposed by Haralick et al. [Har93] (differential matching) for a similar algorithm.

5.2.1 Principle

The two 1-D images to be matched, $l(x)$ and $r(x)$, generally contain the same information (the same objects appear in both images) placed at slightly different positions. Mathematically this could be expressed as

$$r(x) = l(x + d) \tag{5.1}$$

where d is the spatial shift or disparity that has to be recovered, because it is inversely proportional to the distance from the object to the viewer.

For relatively small disparities d, we can apply the Taylor expansion to the right-hand side of equation 5.1 and obtain

$$l(x + d) \approx l(x) + d \cdot \frac{\partial l(x)}{\partial x} = l(x) + d \cdot J(x) \tag{5.2}$$

When plugging equation 5.2 into 5.1 and solving for the disparity we obtain

$$d(x) = \frac{r(x) - l(x)}{J(x)} \tag{5.3}$$

with

$$J(x) = \frac{\partial l(x)}{\partial x} \tag{5.4}$$

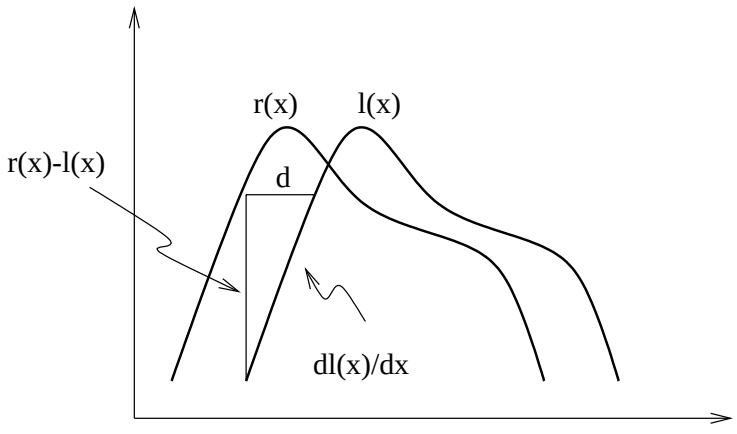


Figure 5.1: A geometric description of equation 5.3: Disparity d can be recovered from the image intensity difference $r(x) - l(x)$ and the derivative of the intensity.

5.2.2 Limitations

Some limitations on the applicability of the differential stereo algorithm stem from equations 5.2 and 5.3.

- The error in the Taylor expansion (eq. 5.2) is proportional to the second derivative of the image intensity. Smoothing the images eliminates higher frequencies and reduces the second-order derivatives. The amount of smoothing will depend on the image content and the desired error level.

  Lucas and Kanade [LUC81] use the inverse of the second derivative as a weight in the disparity calculation. In the case that concerns us, however, disparity is not constant and the second derivative can be only used as a confidence value.

- The Taylor expansion is only valid for small disparities d. Such a condition can be imposed in two ways. The images can be lowpass filtered and subsampled, obtaining an initial coarse value of disparity that is iteratively refined at finer scales [LUC81]. Or else the images can be shifted by a known amount, effectively introducing a known vergence, and the disparity around this new value can be calculated.

- Disparity can not be recovered in constant-intensity areas. When the image derivative $J(x)$ is zero equation 5.3 has no solution. Therefore, disparities measures are more accurate around steeper edges.

- Disparity depends on the difference of image intensities. Therefore a difference in average luminance between images would produce an error in the disparity measure. To eliminate this additive error, images could be bandpass-filtered, so that the average intensity is subtracted.

- A false "zero-disparity" measure appears at pixels i where $l(i) = r(i)$. Disparity can not be solved at those points unless the disparity from neighboring pixels can be used to interpolate a value.

In spite of these limitations, the differential stereo algorithm has two interesting features. It has stereo *hyperacuity*, because the disparity can be measure to subpixel accuracy. And it is a feed-forward calculation that only requires local information: the image intensities and their first (and eventually second) derivatives.

5.3 Experiment

To verify the functioning and the limitations of the differential stereo algorithm, it has been tested in a Matlab interactive program. The input images are smoothed out with a filter of bilateral exponential impulse-response. The derivatives of the left and right images are taken by convolving the smoothed out signals with a $[-1\ 0\ 1]$ mask.

The result of the Matlab simulation is shown in figure 5.2. The upper left corner shows the left and right input images. On the upper right corner the image derivatives and the lowpass versions are shown. The graphic on top shows the derivatives of the lowpass versions and their average, which is actually used to calculate the disparity. The amount of smoothing produced by

the exponential filter is controlled by the `lambda` constant, which can be interactively modified on the lower right section of the image. The disparity, calculated as the ratio of the difference to the average derivative of the smoothed images, appears on the lower left corner of the image.

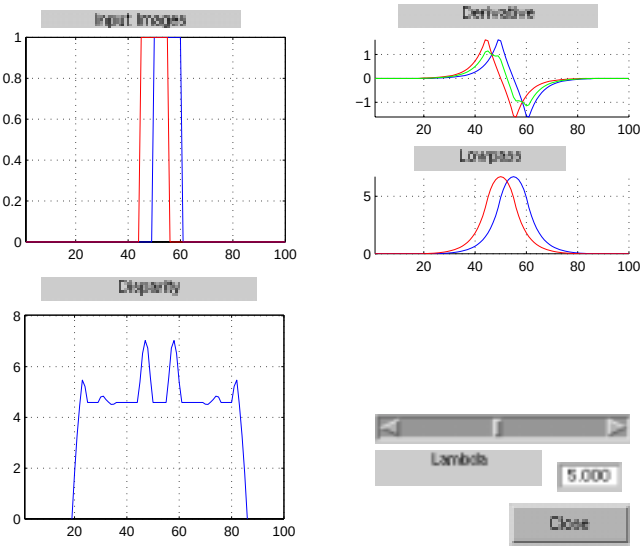


Figure 5.2: Disparity computation on two images shifted by 5 pixels, applying the differential stereo algorithm.

Figure 5.2 shows the disparity calculation for a simple stereo image pair consisting of two pulses of equal width and amplitude with a spatial shift of 5 pixels. As it can be seen, the simple differential stereo algorithm calculates the correct disparity with the exception of some points where the average derivative is smaller. These points produce the central peaks in the disparity image.

Figure 5.3 shows the effects of a gain difference between the left and right image. The left image (the right pulse) has a smaller amplitude than the right one. The disparity is still correctly recovered.

Figure 5.4 shows the ability of the differential stereo algorithm to recover the disparities in the stereo pair when the disparity is not constant. The leftmost pulses have a disparity of 5 pixels, and the right ones have a 3-pixel shift. Except at the transition, disparity is correctly recovered. However, when the two disparity regions are closer (i.e., the *disparity gradient* is greater) the algorithm can not recover the disparity correctly because of the smoothing.

5.4 Circuit

The division in equation 5.3 in a VLSI substrate is difficult. But the problem can be redefined as a multiplication whose error is to be minimized [Tan89] [Bai91]. In our case, equation 5.3

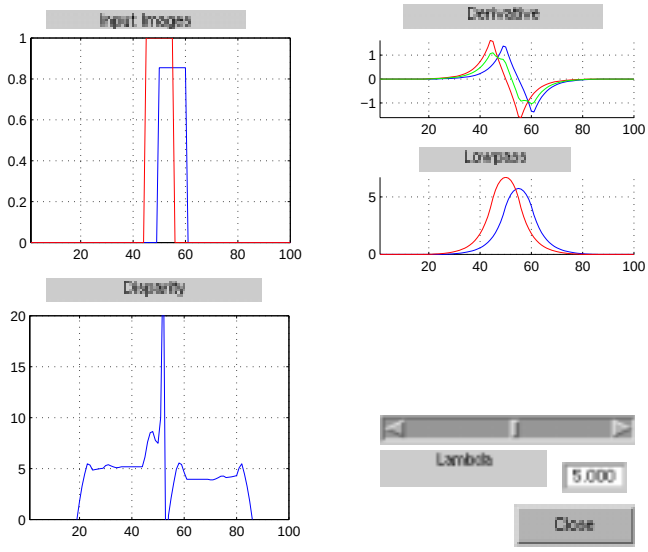


Figure 5.3: Disparity computation on two images of different average intensity, shifted by 5 pixels, applying the differential stereo algorithm. Despite being an intensity-based algorithm, it can recover disparity in images with moderate differences in light intensity.

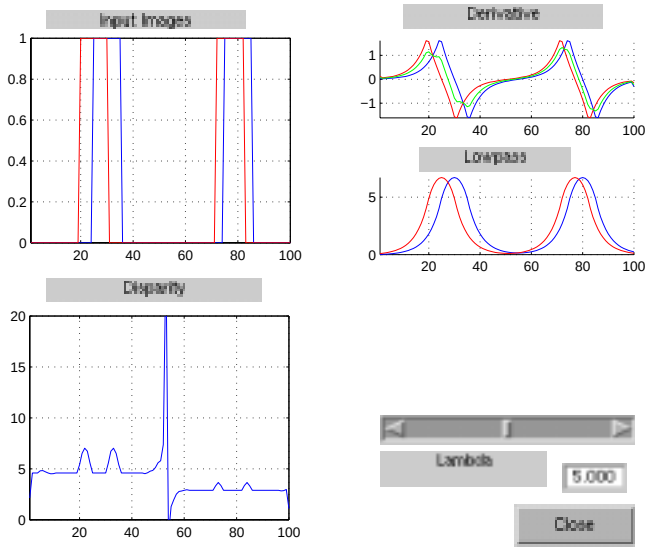


Figure 5.4: Disparity computation on two images with nonuniform disparity. The left pulses have a disparity of 5 and those on the right a disparity of 3.

can be rewritten as

$$[r(x) - l(x)] - d(x) \cdot J(x) = \epsilon \to 0 \tag{5.5}$$

Equation 5.5 is easier to realize, since it involves two subtractions and one multiplication; all of them are simple operation in A-VLSI. Subtraction is carried out in a wire junction, by applying Kirchhoff's current law. Multiplication can implemented by a *transconductance* amplifier, if one of the terms, a current signal, is always positive or zero. In general, disparity $d(x)$ can be negative (*uncrossed disparity*) or positive (*crossed disparity*). $J(x)$, the derivative of the image, can also have negative values, whenever there is a light-to-dark transition.

A *Gilbert multiplier*, described in page 2.4.2, carries out fully-signed multiplication by splitting the signals into two symmetric channels: one for positive signals and one for negative signals. The output signal is obtained by subtracting the two channels [Sto98] [Bis97] [Mea89]. For instance, signal $J(x)$ is decomposed into two channels $J_a(x) > 0$ and $J_b(x) > 0$ according to equation 5.6.

$$J(x) = J_a(x) \; - \; J_b(x) \tag{5.6}$$

This decomposition into symmetric channels fits naturally equation 5.5, because the input images $r(x)$ and $l(x)$, are positive, and their difference $[r(x) - l(x)]$ is a signal that also participates in the equation.

5.4.1 Single cell

By splitting all signals into two channels and using a Gilbert multiplier, equation 5.5 can be solved, i.e., the error is minimized, by the circuit shown in figure 5.5.

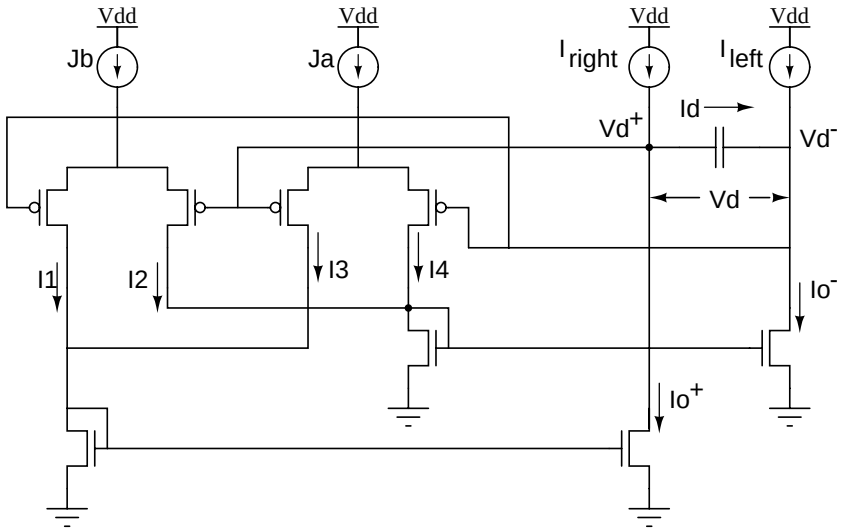


Figure 5.5: Disparity cell: A single cell of the circuit that solves equation 5.5 for each pixel.

Proof The following equations can be identified in circuit 5.5:

- Kirchhoff's equation on the right half of the circuit:

$$\begin{aligned} I_o^+ &= I_{right} - I_d \\ I_o^- &= I_{left} + I_d \end{aligned}$$

- The two current mirrors at the bottom of the circuit:

$$\begin{aligned} I_o^+ &= I_1 + I_3 \\ I_o^- &= I_2 + I_4 \end{aligned}$$

- The two differential pairs

$$\begin{aligned} I_1 - I_2 &= -J_b \cdot k \cdot (V_d^+ - V_d^-) \\ I_3 - I_4 &= J_a \cdot k \cdot (V_d^+ - V_d^-) \end{aligned}$$

with $k = \frac{1}{2nU_T}$ (see section 2.4.2). By operating on the equations above we can obtain

$$\begin{aligned} I_o^+ - I_o^- &= I_{right} - I_{left} - 2I_d \\ (I_1 - I_2) + (I_3 - I_4) &= I_{right} - I_{left} - 2I_d \\ -J_b \cdot k \cdot (V_d^+ - V_d^-) + J_a \cdot k \cdot (V_d^+ - V_d^-) &= I_{right} - I_{left} - 2C\frac{d}{dt}(V_d^+ - V_d^-) \\ (J_a - J_b)k(V_d^+ - V_d^-) &= I_{right} - I_{left} - 2C\frac{d}{dt}(V_d^+ - V_d^-) \\ 2C\frac{d}{dt}(V_d^+ - V_d^-) &= (I_{right} - I_{left}) - (J_a - J_b)k(V_d^+ - V_d^-) \quad (5.7) \\ 2C\frac{dV_d}{dt} &= (I_{right} - I_{left}) - J \cdot k \cdot V_d \quad (5.8) \end{aligned}$$

In steady-state, $\frac{dV_d}{dt} = 0$, and equation 5.8 becomes identical to equation 5.5. The disparity $d(x)$ is coded as the voltage signal V_d.

5.4.2 Circuit inputs

Input signals to the disparity cells are currents that code the filtered intensity at the left and right images ($I_{left}(x)$ and $I_{right}(x)$) and the derivative of the left image $J(x)$. The left and right images have to be bandpass-filtered to both subtract the average intensity and to eliminate the highest spatial frequencies. Further, the left bandpass-filtered image has to be differentiated. All of these operations can be performed on an analog VLSI circuit, further increasing the system integration.

Two resistive networks of different cutoff frequencies can be combined to obtain a bandpass-filtered version of the input currents, in a way similar to the highpass filter discussed in section 2.4.4. Figure 5.6 shows a pixel element of such a bandpass network. The horizontal pseudoresistors controlled by V_{r1} and V_{r2} connect the pixel cell to the neighboring cells. The

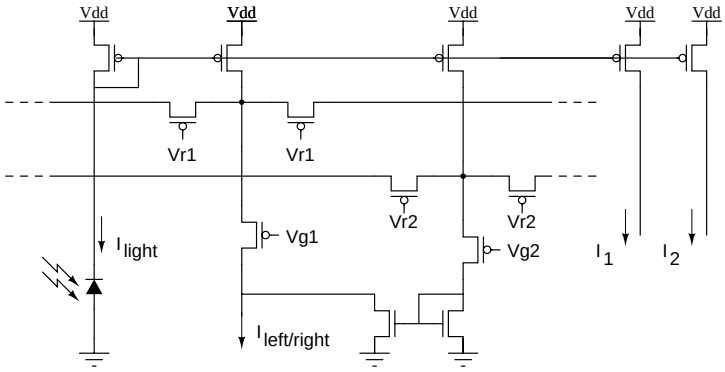


Figure 5.6: A section of a bandpass-filtering resistive network. The photocurrent is smoothed out with two different cutoff frequencies, and the output is the difference of the two filters.

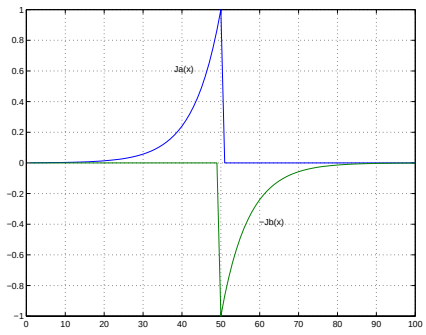


Figure 5.7: Impulse response of two unilateral diffusion networks. Their difference yields the derivative of the lowpass version of the input currents.

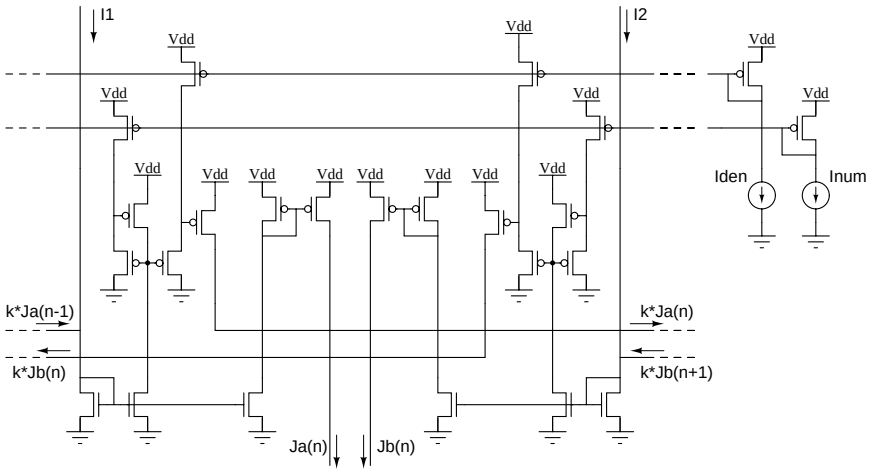


Figure 5.8: Two anisotropic networks produce the derivative of the lowpass of the image. The positive (J_a) and negative (J_b) parts of the derivative can be obtained separately.

ratio $\frac{V_g}{V_r}$ controls the amount of smoothing.

The derivative of an image is more difficult to obtain. But the derivative of the lowpass-filtered version of the image can be computed with special diffusion networks [LAN96]. An unilateral impulse response as shown in figure 5.7 can be obtained by diffusing in only one direction. Figure 5.8 shows an element of the derivative circuit. A fraction of the input current is injected into the neighboring pixels, according to

$$J_a(n) = I_1(n) + k\, J_a(n-1) \tag{5.9}$$
$$J_b(n) = I_2(n) + k\, J_b(n+1) \tag{5.10}$$

where $0 < k < 1$ is the fraction of current injected on the neighboring cell. The Z-transform of equations 5.9 and 5.10 is

$$J_a(z) = \frac{I_1(z)}{1 - k\, z^{-1}} \tag{5.11}$$
$$J_b(z) = \frac{I_2(z)}{1 - k\, z} \tag{5.12}$$

Equations 5.11 and 5.12 show that J_a and J_b are smoothed versions of the input currents $I_1 \approx I_2 \approx I_{light}$, filtered with a convolution kernel the impulse response of which is a unilateral exponential function, as in figure 5.7. The amount of smoothing is controlled by the factor k in equations 5.9 and 5.10. The difference $J(n) = J_a(n) - J_b(n)$ would therefore yield the "derivative of lowpass" version of the input image. In this case however, it is not necessary to subtract them, because the disparity cell (fig. 5.5) requires separate inputs with the negative and the positive parts of the image derivatives.

In the circuit shown in figure 5.8, the fraction $k\,J_i(n)$ that is injected on the neighboring cell is obtained by a translinear loop (cf. section 2.4.2). The variable factor k is controlled by the external —and common to all cells in the derivative circuit— current sources I_{num} and I_{den} according to

$$k = \frac{I_{num}}{I_{den}} \tag{5.13}$$

Therefore, the circuit in figure 5.6 makes two copies (I_1 and I_2) of the photocurrents that are in turn injected into the derivative networks shown in figure 5.8 to produce the currents J_a and J_b that yield the image derivative according to equation 5.6.

5.4.3 Circuit response

Figure 5.9 shows the disparity cell output V_d as a function of the inputs I_{right}, I_{light}, J_a and J_b, as obtained on the `AnaLOG` simulator [Laz86]. The voltage response for constant derivative J is linear in $I_{right} - I_{left}$, as indicated by equation 5.3 and shown in figure 5.9a. Conversely, when the image difference $I_{right} - I_{left}$ is kept constant and the derivative J is varied, the theoretical response is a hyperbola. Figure 5.9b shows the response on the simulator. The saturation that appears for low values of the derivative current J avoids the "division by zero" condition in smooth areas. The error that this saturation introduces can be partially corrected by collective computation among neighboring cells.

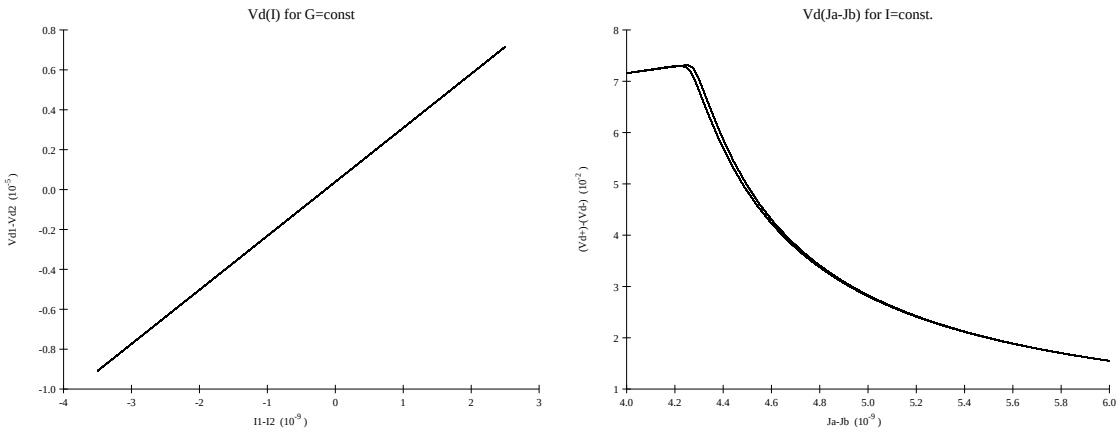


Figure 5.9: Responses of a single cell: (A) Evolution of voltage V_d for a constant derivative $J(x)$, showing the desirable linear response. (B) Evolution of V_d for a constant image difference $I_{right} - I_{left}$, showing an almost hyperbolic response. The saturation for low values of J avoids the "division by zero" condition.

5.4.4 Precision in the differential stereo circuit

As mentioned in chapter 2, the main drawback of analog VLSI circuits is their limited precision, or rather the huge amount of silicon area that is needed to obtain high precision. It is therefore

convenient to estimate the level of precision that could be achieved with the differential stereo circuit, in particular in the disparity cells.

The most important sources of inaccuracy in analog VLSI circuits are transistor parameter mismatch, leakage currents, thermal noise and $1/f$ noise. Thermal noise is produced by the uncoordinated electron motions due to thermal fluctuations, and $1/f$ noise is due to the interface effects between gate oxide and silicon. They both manifest themselves as time-varying, zero-mean, additive errors.

Leakage currents appear as offset errors in the differential stereo circuit. They are space-varying and constant in time.

Transistor mismatch is the error introduced by the fact that transistors do not have exactly the same operating conditions or intrinsic parameters. They manifest as spatially-varying, static errors. Since the differential stereo circuit is made of several disparity cells, there will be a spatial variation between the disparity measures. It is these variations that will be characterized in the next subsection.

Mismatch among transistors can be characterized by the threshold voltage (V_{T0}) mismatch and by the mismatch of the transfer parameter $\beta = \nu C_{ox}\frac{W}{L}$. The threshold voltage mismatch is the dominating factor. In the next subsection, the effects of the threshold voltage variations on the building blocks of the disparity cells will be analyzed, and next section will focus on the total effect of V_{T0} mismatch.

Mismatch in differential pairs

Let it be a differential pair (cf. sec. 2.4.2) where the two transistors have slightly different threshold voltages. The resulting output current would be

$$I_o = I_b \tanh(\frac{v \pm \Delta V_{T1} \pm \Delta V_{T2}}{2nU_T}) \tag{5.14}$$

which for small values of $V = \frac{v}{nU_T}$ can be linearized as

$$I_o \approx I_b \frac{v \pm \Delta V_{T1} \pm \Delta V_{T2}}{2nU_T} = \frac{I_b}{2}(V + \epsilon) \tag{5.15}$$

The error ϵ thus defined has the following statistical properties

$$\begin{aligned} <\epsilon> &= <\frac{\Delta V_{T1}}{nU_T}> + <\frac{\Delta V_{T2}}{nU_T}> = 0 \\ <\epsilon^2> &= \sigma^2(\epsilon) = 2\sigma^2(\frac{\Delta V_T}{nU_T}) \\ \sigma(\epsilon) &= \sqrt{2}\,\sigma(\frac{\Delta V_T}{nU_T}) \end{aligned}$$

Mismatch in current mirrors

If the transistors in a current mirror have different threshold voltages the current mirror equation (eq. 2.9) becomes

$$\frac{I_2}{I_1} = e^{\pm\frac{\Delta V_{T1}}{nU_T}\pm\frac{\Delta V_{T2}}{nU_T}} \approx 1 + \frac{\pm\Delta V_{T1} \pm \Delta V_{T2}}{nU_T} = 1 + \delta \tag{5.16}$$

where δ is a small error with statistics

$$\begin{aligned} <\delta> &= <\frac{\Delta V_{T1}}{nU_T}> + <\frac{\Delta V_{T2}}{nU_T}> = 0 \\ <\delta^2> &= \sigma^2(\delta) = 2\sigma^2(\frac{\Delta V_T}{nU_T}) \\ \sigma(\delta) &= \sqrt{2}\,\sigma(\frac{\Delta V_T}{nU_T}) \end{aligned}$$

Mismatch in the disparity cells

Applying equations 5.15 and 5.16 to the equations in section 5.4.1, describing the functioning of the disparity cell, and supposing a steady-state condition ($I_d = 0$) we obtain

$$I_o^+ = I_{right} \tag{5.17}$$
$$I_o^- = I_{left} \tag{5.18}$$
$$I_o^+ = (I_1 + I_3)(1 + \delta_1) \tag{5.19}$$
$$I_o^- = (I_2 + I_4)(1 + \delta_2) \tag{5.20}$$
$$I_1 - I_2 = -\frac{J_b}{2}(V_d + \epsilon_1) \tag{5.21}$$
$$I_3 - I_4 = \frac{J_a}{2}(V_d + \epsilon_2) \tag{5.22}$$

By adding eqs. 5.21 and 5.22 we obtain

$$(I_1 + I_3) - (I_2 + I_4) = \frac{J_a}{2}(V_d + \epsilon_2) - \frac{J_b}{2}(V_d + \epsilon_1) \tag{5.23}$$

By replacing eqs. 5.19 and 5.20 into equation 5.23 we obtain

$$\begin{aligned} I_{right}(1 - \delta_1) - I_{left}(1 - \delta_2) &= \frac{J_a}{2}(V_d + \epsilon_2) - \frac{J_b}{2}(V_d + \epsilon_1) \\ I_{right} - I_{left} - I_{right}\delta_1 + I_{left}\delta_2 &= \frac{J_a - J_b}{2}V_d + \frac{J_a}{2}\epsilon_2 - \frac{J_b}{2}\epsilon_1 \end{aligned}$$

Isolating the disparity voltage V_d yields

$$V_d = \frac{I_{right} - I_{left}}{\frac{1}{2}(J_a - J_b)} + \frac{-I_{right}\delta_1 + I_{left}\delta_2 - \frac{J_a}{2}\epsilon_2 + \frac{J_a}{2}\epsilon_1}{\frac{1}{2}(J_a - J_b)} = d(1 + e) \tag{5.24}$$

where d is the correct disparity signal $d = \frac{I_{right} - I_{left}}{\frac{1}{2}(J_a - J_b)}$ and e is the relative error $\frac{\Delta d}{d}$ such that

$$e = \frac{-I_{right}\delta_1 + I_{left}\delta_2 - \frac{J_a}{2}\epsilon_2 + \frac{J_b}{2}\epsilon_1}{I_{right} - I_{left}} = \sum a_i x_i \tag{5.25}$$

$$<e> = \sum a_i <x_i> = 0 \tag{5.26}$$

$$<e^2> = \sigma^2(e) = \sum a_i^2 \sigma^2(x_i) = \frac{(I_{right}^2 + I_{left}^2)\sigma^2(\delta) + (\frac{J_a^2+J_b^2}{4})\sigma^2(\epsilon)}{(I_{right} - I_{left})^2} \tag{5.27}$$

$$= \frac{(I_{right}^2 + I_{left}^2)2\sigma^2(\frac{\Delta V_T}{nU_T}) + (\frac{J_a^2+J_b^2}{4})2\sigma^2(\frac{\Delta V_T}{nU_T})}{(I_{right} - I_{left})^2} \tag{5.28}$$

$$= \left(\frac{4(I_{right}^2 + I_{left}^2) + (J_a^2 + J_b^2)}{2(I_{right} - I_{left})^2}\right)\sigma^2(\frac{\Delta V_T}{nU_T}) \tag{5.29}$$

Equation 5.29 shows the effect of the threshold voltage variations $\sigma(\frac{\Delta V_T}{nU_T})$ on the relative error $e = \frac{\Delta d}{d}$ of the disparity voltage V_d. They are related by the constant

$$C = \sqrt{\frac{4(I_{right}^2 + I_{left}^2) + (J_a^2 + J_b^2)}{2(I_{right} - I_{left})^2}} \tag{5.30}$$

Because depth z is inversely proportional to disparity (cf. eq. 3.13), the relative error in disparity and depth measurement is the same:

$$d = d_0(1+e)$$
$$z = \frac{1}{d} = \frac{1}{d_0(1+e)} = \frac{1}{d_0}\frac{1}{1+e} \approx z_0(1-e)$$

Therefore, we obtain the following relationship among depth error, disparity error and threshold voltage variations:

$$\sigma(\frac{\Delta d}{d}) = \sigma(\frac{\Delta z}{z}) = \sigma(e) = C \cdot \sigma(\frac{\Delta V_T}{nU_T}) \tag{5.31}$$

Precision in disparity or depth measurement

The precision obtained in the disparity cell will be evaluated for a test image consisting of a single object which produces a luminance ramp, as in figure 5.10. For this image the following currents would be obtained:

$$L(x) = mx$$
$$R(x) = m(x+d)$$
$$I_{right_{|x_o}} = \alpha\, m(x_o + d) \tag{5.32}$$

$$I_{left_{|x_o}} = \alpha\, m x_o \tag{5.33}$$

$$J_{a_{|x_o}} - J_{b_{|x_o}} = \alpha \frac{\partial L(x_o)}{\partial x} = \alpha m \tag{5.34}$$

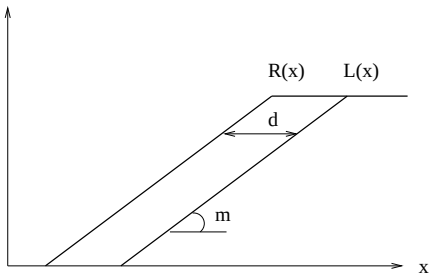


Figure 5.10: Test images, showing a surface of constant luminance gradient m placed at a depth corresponding to disparity d.

Setting the proportionality constant α (cf. eqs. 5.32, 5.33 and 5.34) to a value placing the currents in the subthreshold range, the variance ratio C can be calculated for different values of disparity d. The resulting plot can be seen in figure 5.11.

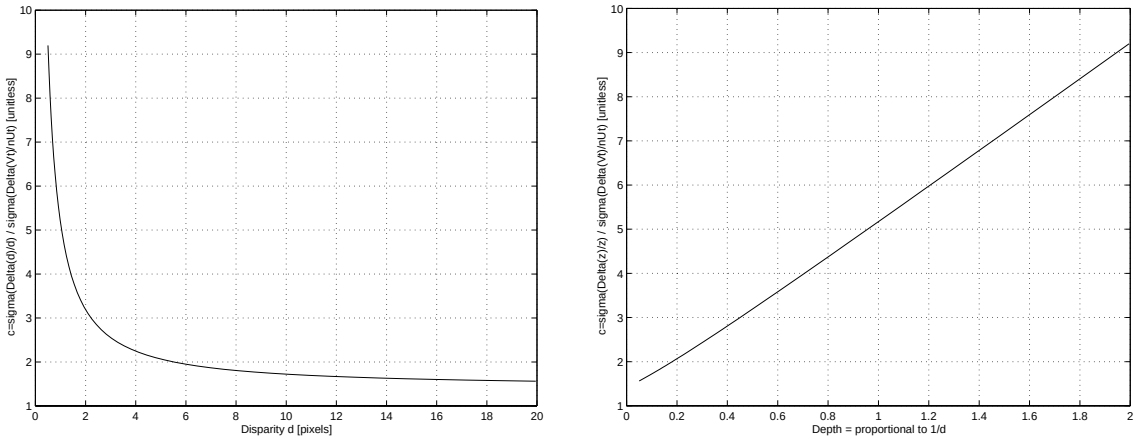


Figure 5.11: The proportionality constant $C = \frac{\sigma(\Delta d/d)}{\sigma(\Delta V_T/nU_T)}$ plotted as a function of the disparity (left) and depth (right), corresponding to the image in figure 5.10.

Figure 5.11 indicates that, for an object producing a disparity of 6 pixels, the constant $C = 2$ and therefore

$$\sigma(\frac{\Delta d}{d}) = \sigma(\frac{\Delta z}{z}) = \sigma(e) = C \cdot \sigma(\frac{\Delta V_T}{nU_T}) = 2\sigma(\frac{\Delta V_T}{nU_T}) \tag{5.35}$$

which means that the standard deviation of the relative error in the depth measurement will be twice as big as the standard deviation in relative threshold voltage $\frac{\Delta V_T}{nU_T}$.

Thus, a technology providing a 2 mV tolerance for the threshold voltage ($\sigma(\Delta V_T) = 2\ mV$), the relative error at a disparity of 6 pixels would be 11%. The actual depth at which this would happen depends on the intrinsic parameters of the vision system, namely the baseline, focal length and pixel pitch. For example, in a system such as the `gemini` turret (see chapter 7) it would take place at 60 cm from the robot.

Discussion

The right half of figure 5.11 plots the evolution of the constant C as a function of $1/d$ which is proportional to the depth. The actual depth depends on the parameters of the vision system (which is independent from the differential stereo circuit), but the plot shows clearly that the relative error $\frac{\Delta z}{z}$ increases linearly with depth.

This section has only analyzed the influence of the mismatches in a single parameter of the transistors: the threshold voltage V_T. Although it is the most important source of errors, other parameters, such as the channel width and length, are subject to similar variations. Parameter mismatches produce constant errors which, in principle, can be eliminated by a calibration procedure.

This calibration, which could be performed on the analog or digital domain, would tune the gain for each pixel based on the measurements on a test image.

There are other noise sources that limit the performance of analog circuits, but these factors have not been considered here, because their influence is smaller. They produce nonconstant errors and therefore cannot be eliminated by calibration. To limit the influence of these noise sources, power consumption and area have to be traded off.

The current analysis has been conducted under the assumption that the input currents are perturbation-free. It must be borne in mind that threshold-voltage mismatches have also a negative effect on the bandpass and derivative circuits. Mismatch produces scaling errors in both pseudoresistors, current mirrors and translinear loops, on which these circuits are based. However the error analysis of these circuits is more complicated because of their collective computation nature.

The accumulation of all these effects may make the circuit useless in the actual form.

5.4.5 Collective computation

Using the circuit in figure 5.5 would simply yield the disparity for each pixel. But a-VLSI offers the possibility of collectively computing the disparity at all pixel sites. Resistive networks and other *collective computation* circuits can be applied to make the disparity at a given pixel dependent on the other pixels in the image.

The *continuity constraint* (cf. section 3.4.1) implies that disparity varies smoothly across the image. A smoothing can be applied on the output voltages V_d by a resistive network, as discussed in section 2.4.4. The individual cells (figure 5.5) are therefore interconnected by resistors

(better implemented as *pseudoconductances*) as shown in figure 5.12.

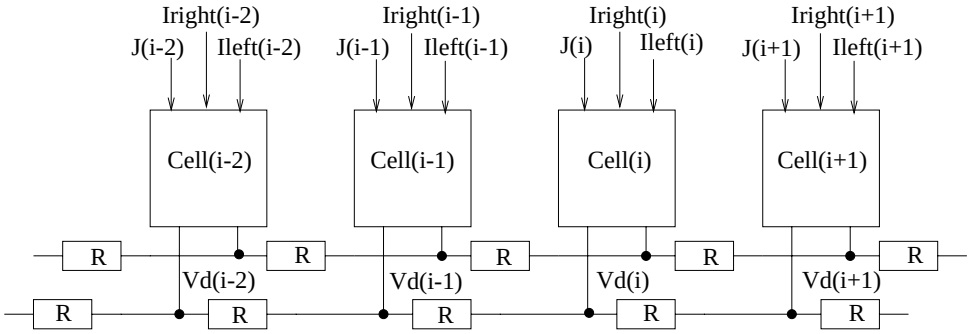


Figure 5.12: Diagram showing the interconnections among the disparity cells (cf. figure 5.5) to obtain a smoothed out disparity measure.

With such an interconnected structure, equation 5.8 becomes

$$2C\frac{dV_d(i)}{dt} = (I_{right} - I_{left}) - J \cdot k \cdot V_d(i) + \frac{1}{R}[V_d(i+1) - 2V_d(i) + V_d(i-1)] \tag{5.36}$$

where R is the pseudoresistance connecting the cells. The solution to equation 5.36 is a function

$$V_d(i,t) = V_{d|_{SS}}(i) + V_{d|_{trans}}(i,t) = V_{d|_{SS}}(i) + A(i)e^{-t/\tau} \tag{5.37}$$

with

$$\tau = \frac{2C}{\frac{2}{R} + J \cdot k} \tag{5.38}$$

$$\tag{5.39}$$

$$V_{d|_{SS}}(i) = \frac{(I_{right}(i) - I_{left}(i)) + \frac{V_d(i+1)+V_d(i-1)}{R}}{J(i) \cdot k + \frac{2}{R}} \tag{5.40}$$

and $A(i)$ depends on the initial conditions.

After a transient time, the disparity voltage $V_d(i,t)$ reaches a steady-state solution (cf. eq. 5.40) that depends on the individual cell's input and its neighbors' output. The resistors R modulate the influence of neighboring cells. When $R \to \infty$ the cells are isolated, and equation 5.40 becomes equation 5.8. In the opposite sense, when $R \to 0$, the terms in eq. 5.40 that do not depend on $\frac{1}{R}$ are negligible and the steady-state value is

$$V_{d|_{SS}}(i) = \frac{V_d(i+1) + V_d(i-1)}{2} \tag{5.41}$$

the average of the two neighbor cells' outputs.

The benefits of connecting the disparity cells with a resistive network are best observed when R has an intermediate value. In that case, if the derivative of the image function $J(\ )$ is zero at a given pixel, the steady-state is kept at a value close to the average of the neighboring outputs

$$V_{d|_{SS}}(i) = \frac{R}{2}(I_{right}(i) - I_{left}(i)) + \frac{V_d(i+1) + V_d(i-1)}{2} \tag{5.42}$$

Similarly, when $I_{right}(i) = I_{left}(i)$ a false "zero-disparity" measure is avoided because of the neighbors' output average in the numerator of equation 5.40.

5.5 Summary

A simple stereo algorithm based on image differences and differentials has been described in this chapter. Under some constraints, the disparity between images can be recovered as the ratio of image differences to their average spatial derivative. The main constraint is that the image be lowpass filtered with a spatial cutoff smaller than the inverse of the smallest disparity.

The difference between the left and right images can be computed on a pixel by pixel basis; the spatial derivative at a given pixel requires only the participation of neighboring pixels. The computation of disparity can take place at a pixel level, because the disparity at a certain pixel (i.e., *cyclopean angle*) depends only on the corresponding left and right pixels and their immediate neighbors. Such an algorithm with a reduced number of interconnections is very convenient for an analog VLSI implementation, because connections are costly (surface is a cost and so is routing) in silicon.

An analog VLSI implementation of the algorithm has been proposed. Resistive networks are used to smooth the images and compute their spatial derivatives. These signals, stemming from the left and right retinas are fed into the disparity cells. These individual cells compute the aforementioned ratio as a multiplication within a feedback loop, bringing the error to a minimum. Furthermore, a resistive network is used to connect the disparity cells to their neighboring cells. In this way, a continuity constraint is imposed on the disparity, adding robustness to the algorithm and avoiding some singular cases in the algorithm.

"Nothing is perfect, said the fox" [SEx91]. The spatial constant of the smoothing filter must be greater than the measured disparity, and too much smoothing hides the possible disparity changes. The disparity gradient that can be resolved is therefore very low. Furthermore, to works properly the algorithm needs images with regions of high contrast, where the spatial derivative is nonzero.

As for the circuit implementation, the error analysis has shown that the overall precision will be severely limited by the mismatch among transistors.

Although the differential stereo can be used with success on robots and artificial systems, there is no proof that the biological mechanism of stereoscopy is based on such simple computation. On the contrary, the aforementioned limitations and psychophysical evidence [Nak85] indicate that stereoscopy (and motion detection) do not rely on the raw image intensities.

Independently of its biological relevance some aspects of a VLSI implementation of the algorithm could be explored. The algorithm's equation (cf. 5.3) is the spatial-domain analogue of

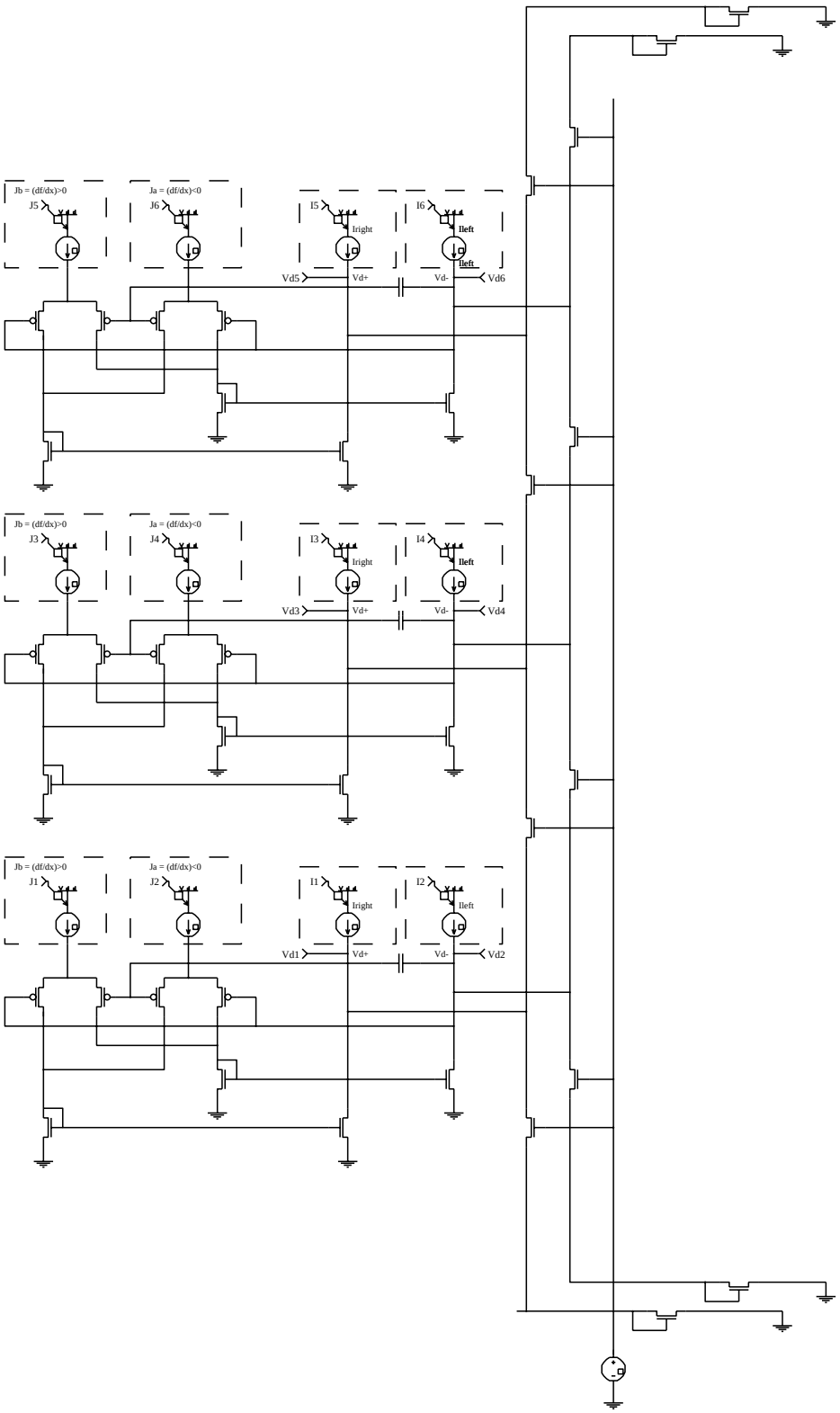


Figure 5.13: Circuit schematic showing three disparity cells (cf. fig. 5.5) interconnected by a network of pseudoresistors.

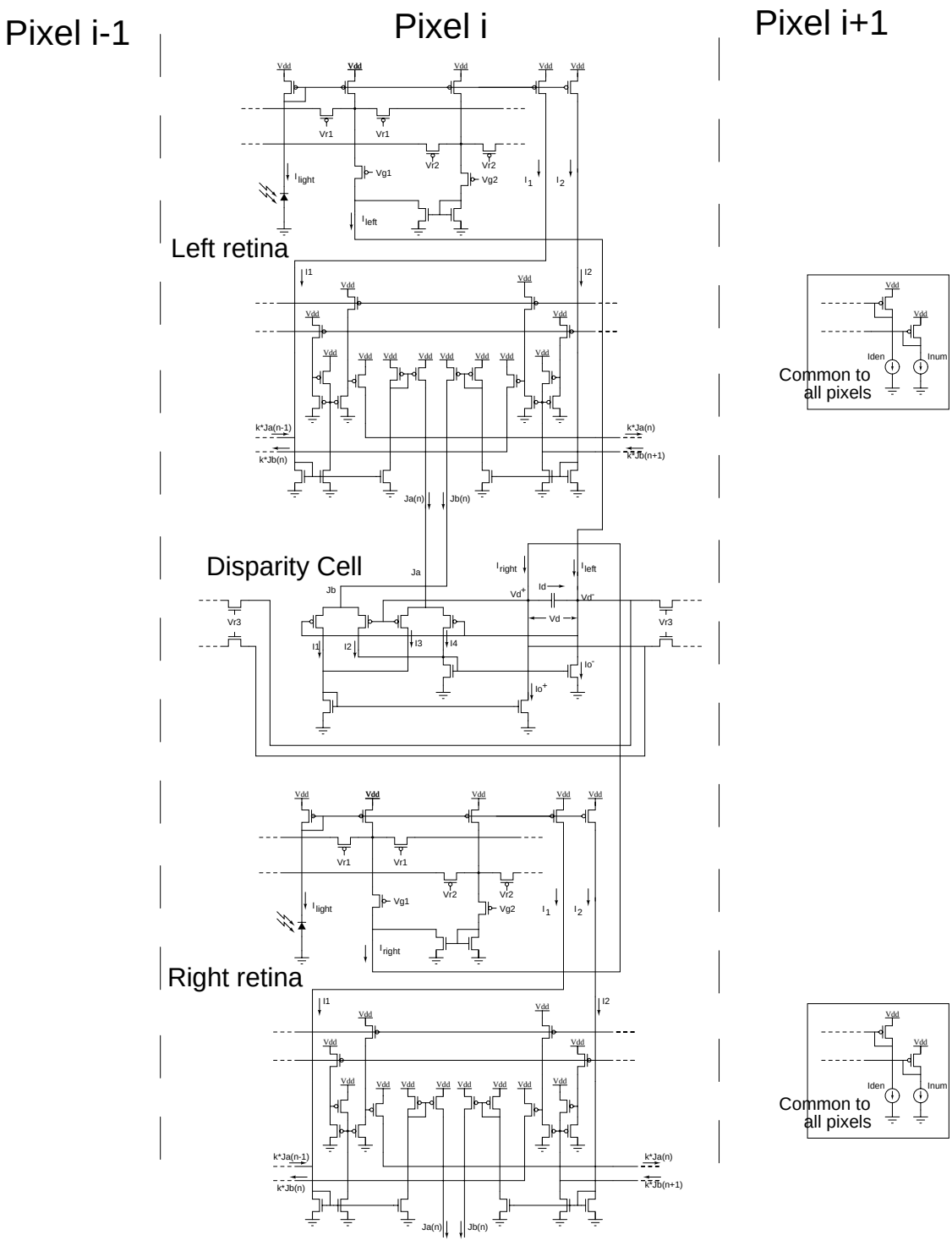


Figure 5.14: Circuit schematic showing the structure for a single pixel. The upper subcircuit is the i^{th} pixel on the left retina, the lower subcircuit is the i^{th} pixel on the right retina, and the central subcircuit is the i^{th} disparity cell. All three subcircuits are connected with their immediate neighbors.

the optical flow constraint equation [Bea95], and several analog VLSI circuits [Sto98] [Tan89] have been proposed for the resolution of the latter equation. An adaptation of these circuits to the spatial domain could provide a more robust implementation of the differential stereo algorithm.

Part III

Hardware and Software Tools for Small-Robot Vision

Chapter 6

A simulated retina

Contents

6.1 Introduction

A simulation of an artificial retina has been developed in the frame of this project. This obeys to two objectives: to gain a better knowledge of the structure and functioning of an artificial retina and to provide a tool for mobile robotics research.

Simulators have always been used in mobile robotics [Mil95] [Gam94] [Mat90] [Ver96] . They allow researchers to develop complex algorithms without having to suffer from unreliable and sometimes expensive hardware. Recently the possibility of working with simple and yet reliable mobile robots, such as the Khepera [Mon94] and Nomad [Nom96], has allowed a greater part of the mobile-robotics research community to switch from simulators to real platforms.

However, some areas still benefit from the use of mobile-robotics simulators. For instance, multiple-partner projects in which the research tasks are distributed among several partners while having a single robotic platform on which to test. In these situations, robot simulators are a practical solution to flexibly test the robot in different environments, but they should be realistic enough to allow the developed algorithms to be used on the real robot. Another area that benefits from robotic simulators is Evolutionary Robotics and Robot Learning. Experiments based on the adaptation capabilities that evolution provides (Genetic Algorithms) or simple learning by an individual robot (Reinforcement Learning) can run for several days or

weeks. That can easily be above the mean time between failures (MTBF) of the hardware and it can be very arduous to supervise by a human if his intervention is needed. Therefore, many works on learning and adaptation have been developed on simulated robots [Ver96],[Gam94].

Mobile-robotics simulators can be roughly classified into two classes which we have defined as processor-based simulators and sensor-based simulators. In the former, the robots usually move in a discrete workspace where the surrounding obstacles occupy cells in the grid [Tom97]. The robot receives inputs from neighboring cells, and these inputs are often of symbolic nature, such as "a person is in the front cell". The processing capabilities of the mobile robot are well simulated, but not the sensory ones. These simulators are very fast and they afford to develop complex algorithms. However, the algorithms are difficult to test on real robots. On the other hand, sensor-based simulators try to mimic the response of the sensors to the surrounding environment [Nom96]. This requires a geometric and physical model of the environment, as well as adequate modeling of the physical processes on which the sensors are based. The simulated robot moves in continuous space, interpreting the sensor signals and deciding the actions to be taken based on this information.

The Webots simulator [Mic98] is a recent example of such sensor-based simulators for mobile robotics. It uses an extension of VRML (Virtual Reality Modeling Language) to store a description of the robot and its environment, and a rendered image of the scene can be obtained by using an OpenGL-compatible graphics library. This in turn allows to simulate vision sensors, which have always been difficult to introduce in robotic simulators. This chapter presents the simulation of a particular vision sensor, the EDI artificial retina, within the Webots simulator. Section 6.2 introduces the artificial retina, and section 6.3 discusses its functional simulation.

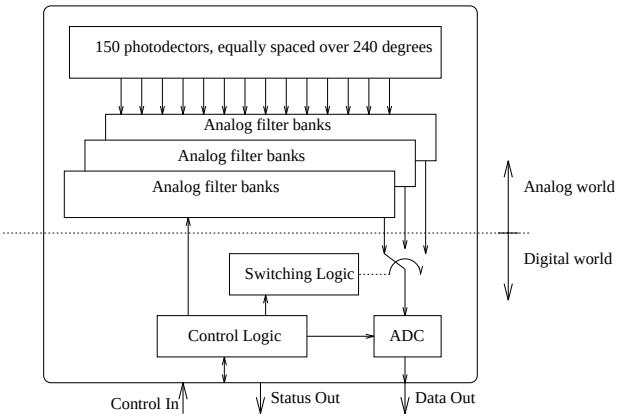
6.2 The Panoramic, Linear, Artificial Retina EDI

The EDI[1] panoramic, linear, artificial retina is an analog-VLSI vision chip designed by Olivier Landolt, at CSEM (Neuchâtel, Switzerland), intended to evaluate the capabilities of such bioinspired sensors. The Microprocessor Laboratory (LAMI) of the EPFL is currently working on the application of such sensors to mobile robotics, and most particularly, on a Khepera robot.

The EDI retina has a linear array of 150 photodiodes lying on an arc that covers 240 degrees of view. But as its name suggests, the EDI is more than a simple vision sensor capable of transducing light. As in most animal retinas, some simple signal processing takes place in the vicinity of the photodetectors, still in the analog domain. This local processing has the effect of filtering the information, reducing its bandwidth, and providing the next stage with a more relevant information. In EDI's case, 3 resistive grids [Vit97] can be combined to apply a lowpass, a bandpass or a highpass filter on the image. An odd-impulse-response filter can also be applied, producing the derivative of the image. The filter output can be accessed by a microprocessor thanks to the internal A/D converter present on the retina. In this way, a 64-gray-level image can be used by the robot for any further processing.

The EDI retina has been mounted on a Khepera turret known as Panoramic turret, together with the necessary ancillary devices and bias circuits. The Panoramic turret has been included

[1] EDI is a shorthand for ED084V2A, the CSEM's device number.



CSEM's artificial retina chip

Figure 6.1: Block diagram of the EDI artificial, linear retina.

as a plug-in module in the current distribution of the Webots simulator [LdM98]. The Steredi turret, described in section 7.3, is a stereo vision turret based on two EDI retinas. The following section discusses the simulation of the EDI retina and the Steredi turret.

6.3 The Steredi Turret on the Webots Simulator

The simulation of the EDI retina follows its functional blocks, which comprise the optics, the photodetectors, the filters and the digital conversion. Each block will be briefly discussed in the subsections that follow. A final subsection will cover the software interface and its compatibility with the real retina.

6.3.1 Optics

The first step of the simulation is to calculate the incident light intensity on each photodetector. We will consider the pixel values of the OpenGL rendering engine as the light intensity falling on the pixel surface. The real EDI retina has a spheric mirror and a lens that focus the image on the photosensors, which lie on the chip as seen in figure 6.2. Ideally, the incident light that reaches the silicon surface should be the same as the incident light that hits a ring on the spheric mirror. Taking this hypothesis, we will try to obtain the incident light on this ring.

The OpenGL rendering engine produces the image of the scene as it would be seen by a flat screen camera. However, the EDI retina is spherical, as it sees an image on 240 degrees in azimuth and 11 degrees in elevation. To solve this problem, on the Panoramic turret [LdM98] the 240-degree view is obtained from two 120-degree (azimuth) by 11 degrees (elevation) projections on a flat surface. For the Steredi turret only 164 degrees are needed for stereoscopic vision, and therefore, to reduce the simulation time, only the 102 pixels that cover this angle are simulated. A 164-degree (azimuth) by 11 degree (elevation) image on a flat surface is projected on a 102 x 7 pixel array.

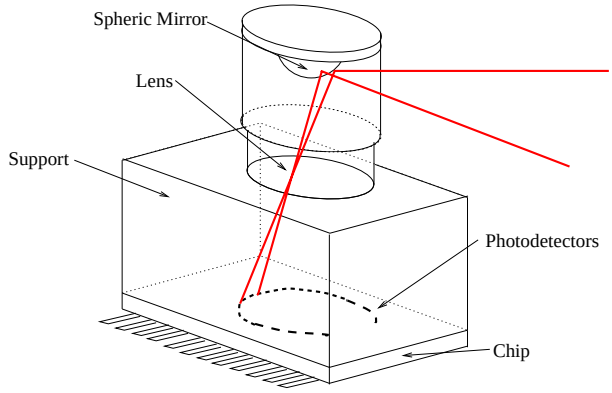


Figure 6.2: The optics of the EDI retina

Next, the array is warped by using the transformations as it can be inferred from figure 6.3. The flat-screen coordinates x,y are transformed into the cylindrical coordinates ϕ and θ as a function of the focal distance f, which is equal to the radius of the cylinder:

$$x = f \cdot \tan(\theta) \Longrightarrow \theta = \arctan(\tfrac{x}{f})$$

$$y = \sqrt{x^2 + f^2} \cdot \tan(\phi) = f \cdot \tan(\phi) \cdot \sqrt{1 + \tan^2(\theta)} \Longrightarrow \phi = \arctan(\tfrac{y}{\sqrt{x^2+f^2}})$$

Finally the vertical pixels are averaged to obtain a 102 x 1 pixel vectors.

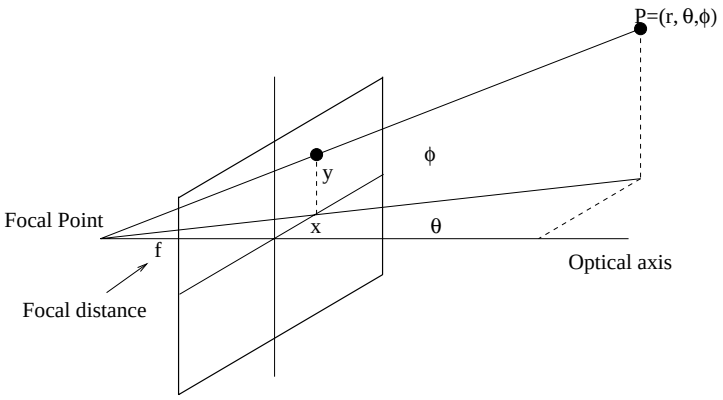


Figure 6.3: Projection of a world in spherical coordinates on to a flat screen

6.3.2 Photodetectors and Normalization Circuit

The image obtained in the previous subsection is considered as a measure of the incident light intensity on each of the 150 photodetectors. The RGB image is converted to gray scale by using the RGB to Y (luminance) transformation of the PAL TV standard. This transformation

gives a certain weight to each color channel. A better modelization could be achieved if the spectral response of the photodetectors were used to generate the luminance. The luminance thus generated is considered as the photocurrent on each photodiode. Next, a normalization circuit divides each pixel's photocurrent by the average current of the whole array. This affords to obtain an illumination-invariant image, that is, an image that will not be perturbed by changes in the global illumination settings of the scene. The normalized current is stored in the model's `analog[ ]` array.

6.3.3 Filtering Layer

The filtering layer simulates the filtering capabilities of the artificial retina. These are implemented with VLSI resistive diffusion networks [VIT97] (cf. sec. 2.4.4) that can be combined to enhance the image and to extract some characteristic features from it. A simple resistive diffusion network has a spatial-lowpass characteristic that can be used to reduce noise in the image and detect uniform regions. Other filters are generated by combining such networks. For instance, the highpass or edge-enhancing filter is obtained by subtracting from the original image its lowpass version. The odd impulse-response filter is generated by subtracting the output from two unidirectional diffusion networks. It produces the derivative of the input image, and it can be used to detect sharp edges in the scene.

1-D resistive diffusion networks behave as low-pass filters of exponential impulse response in the spatial and spatial-frequency domains:

$$h(n) = e^{\frac{-|n|}{\lambda}} \overset{\mathcal{F}}{\Longleftrightarrow} H(\omega) = \frac{\lambda}{\pi} \cdot \frac{1}{1+(\lambda \cdot \omega)^2}$$

Resistive diffusion networks are parallel, locally-connected systems, that perform computations on a real-time basis. This makes it difficult to recreate on a single processor. However, since the system is made out of resistors, that is, linear devices, it can be shown that the output is the convolution of the input signal with a kernel or impulse response of the forms that can be seen in figure 6.4.

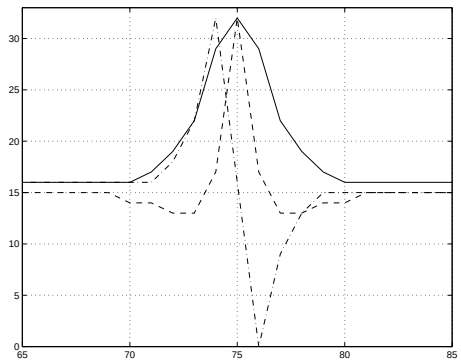


Figure 6.4: Impulse responses of the lowpass (-), highpass (--) and odd (-.) filters.

The result of convolving the `analog[ ]` array with the resistive network impulse response is stored in the `filtered[ ]` array.

6.3.4 Digital Conversion

The EDI retina has a 6-bit successive-approximation A/D converter, that digitizes the output current from the resistive grid into 64 gray levels. Since the analog filters can produce negative values, the higher bit is used as a sign bit. The conversion, defined by the formula $digital[] = 31.5 \cdot (1 + \frac{filtered[\,]}{I_{bda}})$ depends on the variable I_{bda}, which represents a bias current on the EDI chip. In the simulator this variable has been set to a value that makes the output barely saturate when there is a single maximally illuminated photodetector.

6.3.5 The turret

The Steredi turret combines two such simulated EDI retinas. The simulated turret contains the corresponding geometric description of the simulated retinas' position and orientation and the necessary interface calls to access both of them.

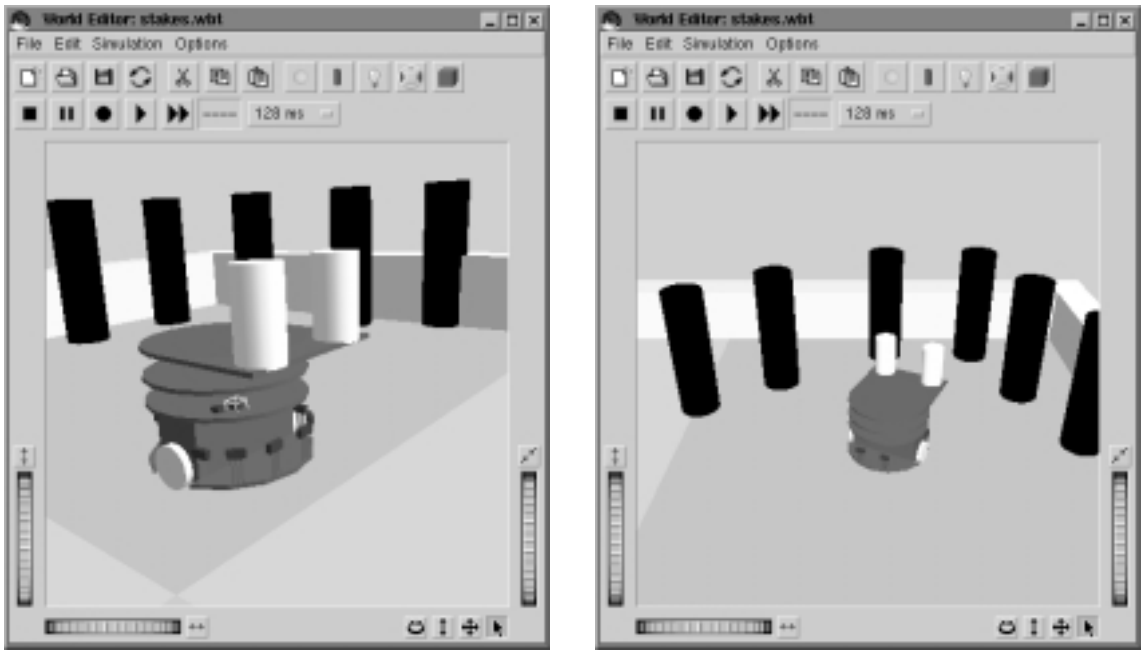


Figure 6.5: (A) View of the Steredi turret and Khepera robot on the Webots simulator. (B) The simulated Khepera facing three black cylinders.

6.3.6 Software Interface: The Khepera API

The benefits of such a sensor-based simulator would be lost if the algorithms developed on a Webots simulated robot cannot be easily applied on a real robot. To that end, a common interface for the real and simulated robot and turrets has been developed, in the form of an Application Program Interface: the Khepera API [API98]. Such a common interface allows the user to use the same C source code, without having to change a single comma, on both platforms. Two different libraries, one for the real robot and one for the virtual one are provided with the We-

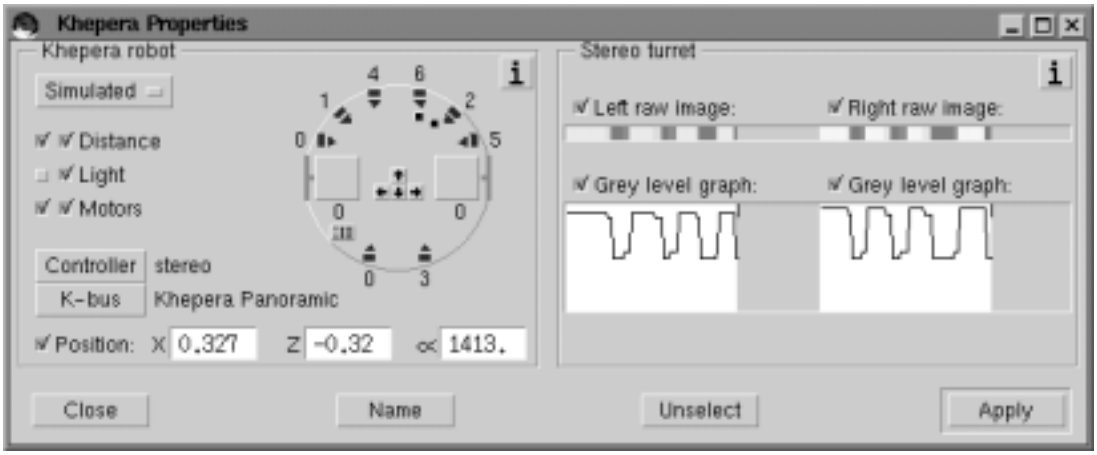


Figure 6.6: Webots window showing, on the right, the output of the simulated Steredi turret and, on the left, the infrared activation levels of the simulated Khepera robot. The Khepera is facing three black cylinders, as shown in figure 6.5b.

bots simulator.

The Khepera API contains several functions to access the Steredi and Panoramic turrets. Some functions allow the user to enable or disable a given filter and other functions are used to read the output of the desired filter from the desired retina. The configuration functions allow the user to set the filter spatial cutoff-frequency and the gray-scale resolution.

6.4 Summary

The modeling of a vision sensor within the frame of the Webots mobile-robotics simulator has been described. Great care has been given to model the physical processes underlying the sensor to obtain a realistic sensor reading. The sensor presented in this chapter is a very particular one, since the EDI artificial retina is a linear, panoramic (i.e., spheric) sensor with signal processing capabilities. However, the same principles can be applied to simulate other vision sensors and standard cameras. The Khepera's commercially available K213 vision turret has been developed in the Webots simulator following the same structure.

The second concern in developing the simulated retina and turret was software compatibility. Simulators are seen as a tool for the time-consuming development of complex algorithms and for sharing data and results among different research teams. In both cases it is highly desirable to be able to test the algorithm on a real robot, and to easily do so, the code should be compatible. For the Webots simulator, an application program interface (API) has been defined, allowing to interface the real and simulated Khepera robot and its turrets with the same C source code.

Chapter 7

Stereoscopic vision turrets for the Khepera robot

Contents

7.1 Introduction

This chapter describes some of the stereo vision systems that have been developed within the frame of this project. The description of the turrets will help the reader to understand the characteristics of some images shown in the previous chapters. The field of view, the resolution and the baseline are important data necessary for the subsequent analysis of the disparity calculations. The reader interested in developing his own miniature vision system may also be interested in learning the decisions that were taken, the components that were chosen and the structure that was used.

The vision systems have been created for the Khepera robot [Mon94], a small, autonomous, mobile robot developed at the LAMI-EPFL and commercialized by K-TEAM SA, Lausanne, Switzerland. This small robot of 55 cm in diameter and 4 cm in height is equipped with two wheels, 8 infrared sensors and a Motorola 68331 microprocessor. One of the key features of the Khepera robot is its extensibility; new modules, called "turrets" in the Khepera jargon, can be plugged on top of the robot and connected with the Khepera's processor via a serial or a parallel bus.

7.2 The Gemini Turret

The `gemini` turret is a digital vision system based on two linear sensors and two parallel A/D converters. It is plugged on top of the 68331 universal extension turret (`ext331` turret) of the

Khepera robot, so that the turret's microprocessor interfaces the converter through its data lines, as shown in the diagram 7.1.

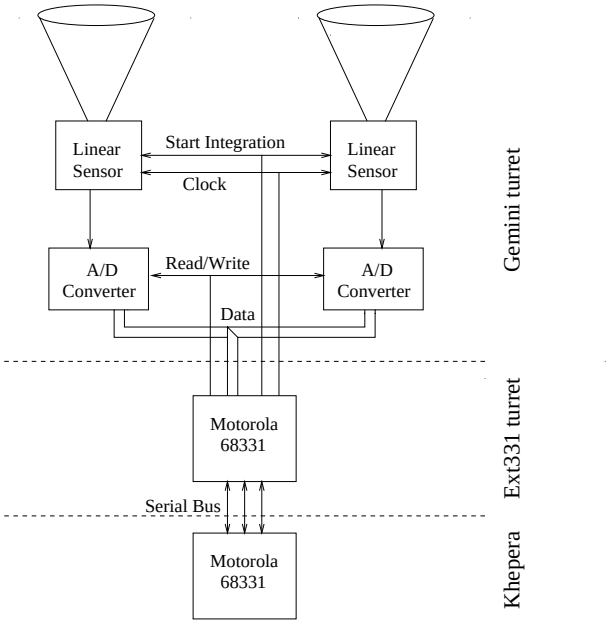


Figure 7.1: Structure of the `gemini` turret, mounted on the `ext331` turret of the Khepera robot.

7.2.1 Optics and sensing

The optics for the `gemini` turret have been fabricated at the Microprocessor Systems Lab (LAMI) of the EPFL. Two plastic blocks protect the sensors from extraneous light and hold the spheric lenses on a screw thread so that they can be focused.

Figure 7.2 shows a picture of the `gemini` turret; the black cylinders on the vertical circuit board contain the optic system and the sensors. The circuit below it contains the two A/D converters and miscellaneous glue logic. The circuit just below, sitting on top of the Khepera is the `ext331` turret.

The sensors that have been chosen are Texas Instruments TSL 1301 Linear Sensor Arrays. Each array consists of 102 photodiodes that feed a charge amplifier. After a certain integration time, the charge amplifiers are sequentially read, obtaining an analog voltage between 0 and 2.2 volts.

These sensors have been used because of their low cost, relatively good resolution and their ease of interface. Only 2 inputs lines are needed (one for the "start integration" pulses, and another for the "next pixel" pulses). The analog image intensities are sequentially output on a single line.

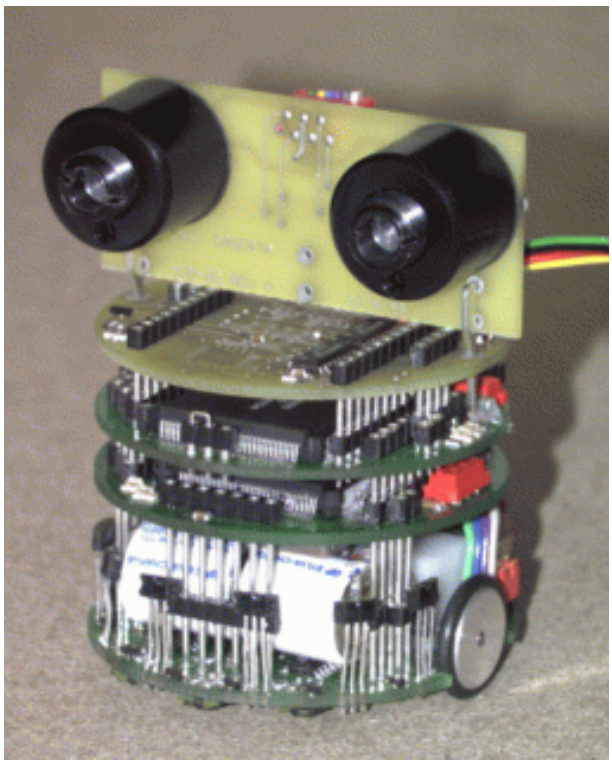


Figure 7.2: The `gemini` turret, mounted on the `ext331` turret of the Khepera robot.

7.2.2 Conversion

The output voltage from each sensor is sent to two 8-bit, parallel ADC 0820 A/D converters[1]. The converters are controlled by the microprocessor, using the RD-WR mode of the ADC 0820. In this mode, the microprocessor signals the beginning of conversion by issuing a WR pulse and reads the digital output 1.4 μs later by issuing a RD pulse.

The images are stored on the `ext331` turret's memory, and they can be further processed (feature extraction, stereo matching) or sent via serial line to a workstation to be displayed (see figure 4.11).

Optics	
Vergence angle	0
Field of view	$\approx$ 57 deg.
Focal length	7 mm
Baseline	43 mm
Sensing	
Sensor type	Photodiodes
Number of pixels	2 x 102
Resolution (pitch)	85 μm
Integration time	min. 8 ms
Processing	None
Conversion	
Access	sequential
A/D conversion	parallel
Gray levels	256 (8 bits)
Maximum level	216
Scan rate	max. 120 Hz

Table 7.1: System characteristics of the `gemini` turret.

7.3 The Steredi Turret

The `steredi` turret is based on two analog, artificial retinas EDI, developed by the Swiss Center for Electronics and Microtechnics, (CSEM SA), Neuchâtel, Switzerland. These retinas, developed by Olivier Landolt, are described in section 6.2. The pages that follow will describe their use in a stereo vision turret, the `steredi` turret, shown in figure 7.3.

The EDI artificial retinas used in the turret are equipped with their own optic systems (cf. section 6.2) that allow them to have a linear, panoramic view of their environment. The retinas themselves perform the image sensing, normalization, processing and A/D conversion.

[1]The converters can be obtained from different sources under the common code ADC 0820.

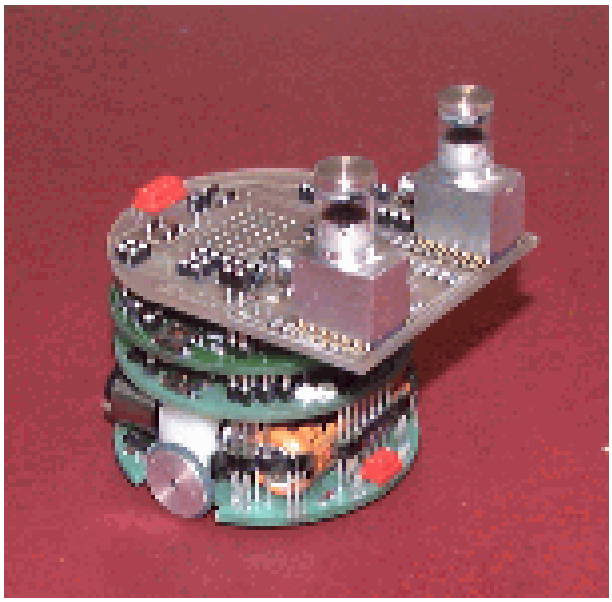


Figure 7.3: The `steredi` turret, containing two EDI artificial retinas, mounted on top of an `ext331` turret on the Khepera robot.

The EDI retinas have 150 pixels covering 240 degrees, but for stereo applications the stereo field of view, that is, the field that is covered by both eyes, is limited to 164 degrees or equivalently, 102 pixels (see figure 7.4. Therefore, when reading, the first (leftmost) 24 pixels are skipped and only the following 102 pixels are read.

The `steredi` turret is mounted on the `ext331` turret of the Khepera robot. Four microprocessor data lines per retina are used to configure them. A lowpass filter, bandpass filter or derivative filter can be selected, and their cutoff frequencies set. The retinas contain a 6-bit sequential A/D converter; one of the data lines is used to read out the pixels bit by bit.

The block structure of the `steredi` turret is shown in figure 7.5.

The turret contains other components that are necessary for the correct functioning of the EDI retinas. Several resistors running into the megaohms generate the necessary bias currents that control several internal parameters of the retina. Only one of these parameters, the A/D converter reference current, can be modified at run time. To do so, two LTC 1257 D/A converters have been added to the turret, allowing the microprocessor to modify a reference voltage by sending a 12-bit stream to the converter. A detailed description of the bias currents appears in [Lan96].

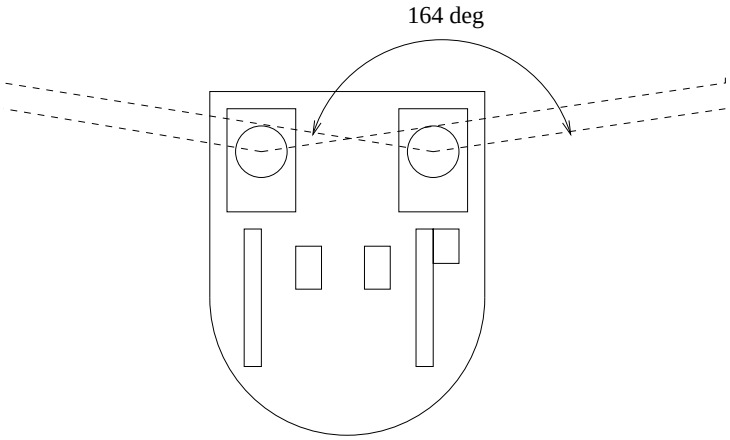


Figure 7.4: Top view of the `steredi` turret, showing the stereoscopic field of view.

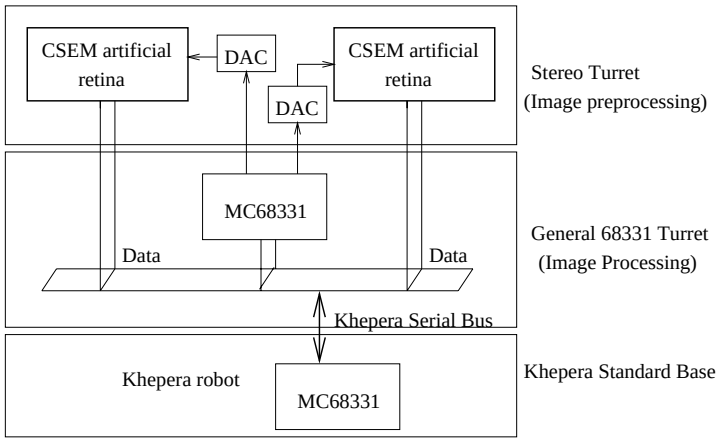


Figure 7.5: Block diagram of the `steredi` turret.

Optics	
Vergence angle	0
Field of view	164 deg.
Focal length	N/A
Baseline	36.1 mm
Sensing	
Sensor type	Photodiodes
Number of pixels	2 x 102
Resolution (pitch)	1.6 deg.
Processing	Lowpass, bandpass, derivative
Conversion	
Access	sequential
A/D conversion	serial
Gray levels	64 (6 bits)
Scan rate	—

Table 7.2: System characteristics of the `steredi` turret.

Chapter 8

Conclusion

This dissertation addresses the problem of endowing small, autonomous, mobile robots with depth perception. Although depth perception can be obtained by other means —motion parallax, depth from focus, cognitive cues— stereoscopy is found to be the most robust and adequate for such robots.

Analog VLSI circuits can advantageously solve some sensory problems for autonomous, mobile robots. The advantages stem from the sensing and processing integration, real-time computation, limited power consumption and small size of analog VLSI circuits.

An analysis of several analog VLSI circuits has identified the principles and basic elements of analog VLSI circuits. Algorithms that are easy to implement in analog VLSI have to satisfy two major conditions. First, they must be decomposed into numerous, parallel, simple, computation units. Although the computational units are simple, highly complex and nonlinear behaviors can still be obtained by establishing interactions among units and between units and common signals. Second, the intercommunication among these units must be limited. It should be noted that these two conditions can also benefit sequential, digital implementations of the algorithm, because they simplify the programming and reduce the memory required.

Two stereoscopic algorithms have been proposed to solve this necessary task in autonomous, mobile robot navigation: depth perception. Several stereoscopic algorithms have been proposed by Computer Vision researchers and implemented in analog VLSI. However this dissertation concentrates on developing even simpler algorithms that follow the aforementioned conditions. They are based on simple elementary computations that only require local information stemming from the neighboring computation cells.

The first algorithm that has been discussed exploits the phase properties of the Gábor functions to recover the disparity in a stereo pair of images. Disparity can be computed at each position based on information from the corresponding retinal positions and their immediate neighbors. Such a structure affords a decomposition into multiple, parallel units that simplify an analog VLSI implementation as well as a digital one. A contribution has been made by proposing two approximations, based on a Gaussian and exponential expansion that further simplify the application on analog and digital systems.

Emphasis has been put into solving an engineering problem —providing a small, mobile robot with depth perception by stereo vision— rather than modeling the functioning of the brain. Although the Gábor function has been found to model the receptive field of some cells in the visual cortex, it is not likely that the brain computes disparity as in this algorithm. More

biologically-plausible algorithms could be obtained based on the pseudogabor function.

A second stereo algorithm, also based on local, differential computations, has been analyzed and tested. An analog VLSI circuit exploits this local processing, similar to optical flow computation, to solve the disparity. In spite of its inherent limitations, the algorithm is found to be robust and efficient in constrained images.

The overall contribution of this dissertation is in focusing on a mobile robotics problem and propose solutions that can be implemented in both analog VLSI circuits —bringing forth the advantages of this medium— and simple, digital systems.

8.1 Towards an analog VLSI implementation

The problems and solutions discussed in this dissertation open a number of questions that should be addressed before actually implementing the algorithms in analog VLSI. The study of these questions should improve the circuit implementation and system design.

Stereoscopic vision raises the question of communication between analog circuits. The *baseline distance* of the vision system has to be sufficiently long to have a good and accurate depth perception, making it very difficult to integrate both retinas on the same silicon die. It is therefore necessary to provide a means of communicating between retinas (if the stereo matching takes place on the retinas) or between retinas and a stereo matching chip. Establishing a wire connection for each pixel is out of question because of lack of space. Therefore some kind of multiplexing will be necessary.

One solution could be to digitize the visual information and use a microprocessor to send the data to the stereo chip. Since most robots use digital microprocessors this is a reasonable solution, but some of the advantages of analog VLSI circuits would be lost.

Another solution would be the use of an asynchronous, analog, pulse-based communication scheme as proposed in [Mah92] and [Mor94]. Such a solution would require the use of retinas and stereomatching chips based on the *Address-Event Representation*, an analog pulse-coding technique. The AER coding would simplify the introduction of new features such as software vergence —an artificial shift in the images— and multiscale processing —processing the image at several resolution levels.

8.2 Future work

A subject to be explored is the deeper integration of stereoscopic vision into the sensorimotor loop. Mobile robots are not static platforms; stereo vision algorithms can benefit from the mobility of the platform on which they are mounted, particularly to solve mismatches due to occlusions.

One way to realize this integration is through **active stereovision**, the modification of the parameters that control the vision mechanisms, based on the information from the image, the current goal of the vision system and the motion of the platform. In the stereo algorithms

described in this dissertation the parameters that can be modified are the modulating frequency and the amount of smoothing, but others —vergence, fixation point— could be added. Analog VLSI offers a good medium to realize active vision, because it has proved to be very effective in integrating sensors, processors and even actuators such as the micro electromechanical systems (MEMS).

In the near future, autonomous mobile robots will become a common item in our lives. They are already the spearhead of planetary exploration and they are becoming household items such as maintenance robots and toys. Tight sensorimotor loops based on bioinspired, smart sensors will offer these robots with a greater energetic and decisional autonomy. Conversely, experimentation with robotic sensorimotor loops will allow biologists to gain insights on the sensorimotor mechanisms found in animals.

Appendices

Calculation of the horopter

In the situation described in figure 3.2 the coordinates of the point of interest P was

$$x = \frac{B}{2} \cdot \frac{\tan(\alpha+\beta) + \tan(\gamma+\delta)}{\tan(\alpha+\beta) - \tan(\gamma+\delta)} \tag{1}$$

$$z = \frac{B}{\tan(\alpha+\beta) - \tan(\gamma+\delta)} \tag{2}$$

The latter can be rewritten as

$$z = \frac{B}{2} \cdot \frac{\cos(\alpha+\beta+\gamma+\delta) + \cos(\alpha+\beta-\gamma-\delta)}{\sin(\alpha+\beta-\gamma-\delta)} \tag{3}$$

and equation 1 as

$$x = \frac{B}{2} \cdot \frac{\sin(\alpha+\beta+\gamma+\delta)}{\sin(\alpha+\beta-\gamma-\delta)} \tag{4}$$

Defining the variable $\phi = \alpha - \gamma$, the vergence angle, and an auxiliary one $\rho = \alpha+\beta+\gamma+\delta$, equation 1 becomes

$$x = \frac{B}{2} \cdot \frac{\sin(\rho)}{\sin(\phi)} \tag{5}$$

and equation 3 becomes

$$z = \frac{B}{2} \cdot \frac{\cos(\rho) + \cos(\phi)}{\sin(\phi)} \tag{6}$$

$$z = \frac{B}{2} \cdot \frac{\cos(\rho)}{\sin(\phi)} + \frac{B}{2} \cdot \frac{\cos(\phi)}{\sin(\phi)} \longrightarrow z - \frac{B}{2} \cdot \frac{\cos(\phi)}{\sin(\phi)} = \frac{B}{2} \cdot \frac{\cos(\rho)}{\sin(\phi)} \tag{7}$$

From equations 7 and 5 we have that

$$\left[z - \frac{B}{2\tan(\phi)}\right]^2 + x^2 = \left(\frac{B}{2\sin(\phi)}\right)^2 \tag{8}$$

which is the equation of a circle that depends only on the vergence angle ϕ. The cercle has a radius

$$r(\phi) = \frac{B}{2\sin(\phi)} \tag{9}$$

and its center is at

$$\vec{c}(\phi) = (0, \frac{B}{2\tan(\phi)}) \tag{10}$$

Approximation of a Gaussian function by an exponential one

In order to approximate a Gaussian function

$$g(x,\sigma) = e^{-\frac{x^2}{2\sigma^2}} \tag{11}$$

by a bilateral decaying exponential

$$f(x,\lambda) = e^{-\frac{|x|}{\lambda}} \tag{12}$$

it is necessary to know what is the λ that produces the best approximation in the least-squares sense for a given σ.

We therefore define the approximation error as

$$\begin{aligned} E &= \int_{-\infty}^{\infty} [g(x,\sigma) - f(x,\lambda)]^2\, dx = \int_{-\infty}^{\infty} [g^2(x,\sigma) - 2g(x,\sigma)f(x,\lambda) - f^2(x,\lambda)]\, dx && (13)\\ &= \int_{-\infty}^{\infty} e^{-\frac{2x^2}{2\sigma^2}}\, dx - 2\int_{-\infty}^{\infty} e^{-\frac{x^2}{2\sigma^2}} e^{-\frac{|x|}{\lambda}}\, dx + \int_{-\infty}^{\infty} e^{-\frac{2|x|}{\lambda}}\, dx && (14)\\ &= A + B + C && (15) \end{aligned}$$

with

$$\begin{aligned} A &= 2\int_0^{\infty} e^{-\frac{x^2}{\sigma^2}}\, dx = 2\frac{\sqrt{\pi}}{2}\sigma\, erf(\infty) = \sqrt{\pi}\,\sigma\\ C &= 2\int_0^{\infty} e^{-\frac{2x}{\lambda}}\, dx = \lambda && (16) \end{aligned}$$

and

$$B = -4\int_0^{\infty} e^{-\frac{x^2}{2\sigma^2}} e^{-\frac{x}{\lambda}}\, dx \tag{17}$$

which can be found in table books [GRA94] as

$$\int_u^{\infty} e^{-\frac{x^2}{4\beta} - \gamma x}\, dx = \sqrt{\pi\beta} e^{\beta\gamma^2}[1 - erf(\gamma\sqrt{\beta} + \frac{u}{2\sqrt{\beta}})] \tag{18}$$

with $erf(x)$ defined as

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt \tag{19}$$

In the case of equation 17 the constants are

$$\begin{aligned} \beta &= \tfrac{\sigma^2}{2} \\ u &= 0 \qquad (20) \\ \gamma &= \frac{1}{\lambda} \qquad (21) \end{aligned}$$

and therefore equation 17 can be rewritten as

$$B = -2\sqrt{2\pi}\sigma e^{\frac{\sigma^2}{2\lambda^2}}[1 - erf(\frac{\sigma}{\lambda\sqrt{2}})] \tag{22}$$

Therefore the approximation error is

$$E = A + C + B = \lambda + \sqrt{\pi}\sigma - 2\sqrt{2\pi}\sigma e^{\frac{\sigma^2}{2\lambda^2}}[1 - erf(\frac{\sigma}{\lambda\sqrt{2}})] \tag{23}$$

The λ that minimizes the error is such that

$$\begin{aligned} 0 = \frac{\partial E}{\partial \lambda} &= 1 - \sqrt{8\pi}\sigma\{\frac{-2\sigma^2}{2\lambda^3} e^{\frac{\sigma^2}{2\lambda^2}}[1 - erf(\frac{\sigma}{\lambda\sqrt{2}})] + e^{\frac{\sigma^2}{2\lambda^2}}(\frac{-2}{\sqrt{\pi}}) e^{-\frac{\sigma^2}{2\lambda^2}}(\frac{\sigma}{\sqrt{2}})(\frac{-1}{\lambda^2})\} \qquad (24) \\ &= 1 + \sqrt{8\pi}\frac{\sigma^3}{\lambda^3} e^{\frac{\sigma^2}{2\lambda^2}}[1 - erf(\frac{\sigma}{\lambda\sqrt{2}})] - 4\frac{\sigma^2}{\lambda^2} \qquad (25) \end{aligned}$$

By defining the ratio $n = \frac{\sigma}{\lambda}$ which is the ratio we are actually looking for, equation 25 can be written as

$$1 + \sqrt{8\pi}\, n^3 e^{\frac{n^2}{2}}[1 - erf(\frac{n}{\sqrt{2}})] - 4n^2 = 0 \tag{26}$$

Equation 26 is nonlinear and therefore an analytic solution can not be found. Nonetheless it can be solved by numerical means or plotted (see figure 1) and the solution is found to be $n_{opt} = 0.762$.

Therefore, the best approximation of the Gaussian function of variance σ with a bilateral, exponentially-decaying function of decay constant λ is the one where

$$\lambda = \frac{\sigma}{n_{opt}} = 1.312\,\sigma \tag{27}$$

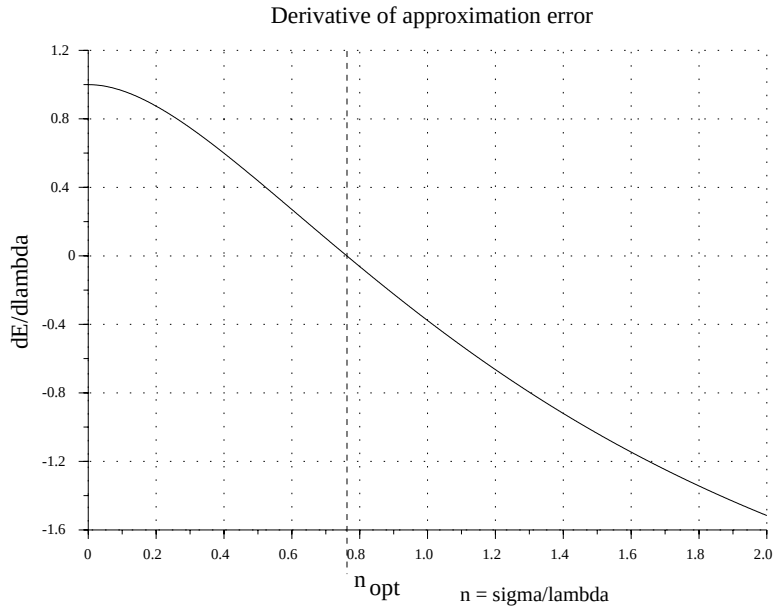


Figure 1: The optimum $\frac{\sigma}{\lambda}$ ratio is found to be $n_{opt} = 0.762$.

Glossary

A-VLSI: Analog, Very Large Scale Integrated circuits: Continuous-valued electronic circuits manufactured in Very Large Scale Integration (VLSI) technology. This technology affords the integration of about 50,000 transistors (elementary computation units) per mm^2.

Above threshold: A CMOS transistor operating region, also called *strong inversion*. In this region the currents flowing into and out of the transistor follow a square law.

Acuity: The reciprocal of the minimum angular separation between two points that can be resolved by a vision system. *Vernier acuity* refers to the resolvable difference in angle or retinal position and stereoacuity refers to the resolvable difference in *depth*. Often expressed in angle units or linear units on the focal plane. See *stereoacuity*.

Address-Event Representation: An analog VLSI signal coding technique whereby each element or 'cell' reacts to a certain event (e.g., the passage of an edge in front of a photodetector) by asynchronously sending its address on a common bus with a firing rate proportional to the intensity of the event. Such technique affords a bandpass reduction, simplifies interchip communications, allows data multiplexing and dynamic routing and facilitates multiscale processing.

APS: Active Pixel Sensor: a CMOS photosensor with in-pixel current amplifiers, as opposed to standard CMOS sensors, where the output amplifier is shared among the pixel columns. See *CMOS*.

Area-based methods: Image Processing algorithms where the information to be processed or analyzed is obtained from an area around a given pixel instead of the pixel itself.

Baseline distance: In stereoscopy it is the distance separating the two eyes or image sensors, usually measured from the point where the optic axis intersects the focal plane, i.e., the retinal position (0,0).

CCD: Charge-Coupled Device: The most common photosensor technology, whereby the phototransduced charges are shifted through the pixel lines to a charge-sensitive amplifier.

CDS: Correlated Double-Sampling: An acquisition technique whereby the first sample (takes after the photodetector reset) is subtracted from the final sample (after integration), thereby eliminating the DC offset and reset noise.

Center on-surround off: A neural response whereby stimuli in the center of the receptive field increase the neuron activity (excitatory connection) and those in the periphery

decrease the neuron activity (inhibitory response). The opposite structure is the center-off-surround-on.

CMD: Charge-Modulated Device: A hybrid technology between CCD and CMOS photosensors that produces a high fill-factor with low dark current. It is a type of APS. See *APS, CCD.*

CMOS: Complementary Metal-Oxide-Semiconductor: An integrated-circuit fabrication technology whereby transistors devices are obtained at the interfaces of 3 layers of metal, insulator (silicon dioxide) and semiconductor. This process has a particularly advantageous power consumption and it has become the most widely available. A variant of the CMOS process is used to produce photosensor arrays, and these sensors are referred to as CMOS photosensors.

Coherence principle: A weaker version of the continuity constraint in stereoscopic vision. Proposed by Prazdny, it states that a disparity discontinuity may appear as a result of the superposition of several continuous disparities, e.g., in the presence of transparent surfaces. See *continuity constraint.*

Collective computation: An analog computation technique whereby the result is calculated by continuous interaction between simple, elementary computational units.

Compatibility constraint: In stereoscopic vision this constraint states that features in one image can only match the same class of features in the other image.

Continuity constraint: In stereoscopic vision this constraint states that the resulting disparity must be a continuous function, except at the object boundaries.

Contralateral: In systems with a symmetric structure such as the human body, it refers to connections coming from the opposite side. For instance, contralateral connections bring the nerves of the left hand to the right hemisphere of the brain. Opposite: *ipsilateral.*

Cooperative stereo: A class of stereovision algorithms whereby the resulting disparity is computed by cooperation among individual computation units, each responsible for a single disparity at a given position.

Correspondence problem: In stereoscopic vision, it is the process of identifying the projection on the left image of certain points (features) in the scene and their corresponding projections on the right image.

Crossed disparity: In stereoscopy, when in order to fixate on a point the vergence has to be increased, i.e., the optic axes need to be further crossed, the given point is said to be at a crossed disparity with respect to the current fixation point, that is closer to the observer. See *fixation point, horopter.*

Current conveyor: An analog circuit that copies (conveys) the input current to the output node therefore isolating the output voltage from the input one.

Current mirror: A two-transistor analog circuit that produces an output current proportional to the input current. The input and output voltages are not independent as in the *current conveyor.*

Current mode: A class of circuits, analog or digital, where signals are coded as currents.

Cyclopean angle: In stereoscopy, the angle under which an object would be seen by a virtual eye or camera placed at the center of the baseline with its optic axis aligned with the baseline median. See *baseline*.

Decussation: In anatomy, an X-shaped crossing especially of nerve fibers connecting corresponding parts on opposite sides of the brain or spinal chord.

Depth: In stereoscopy the depth of a given point is the minimum distance from that point to the baseline, the line between the two eyes or cameras. It is the projection of the object distance along an axis perpendicular to the baseline. See *baseline*.

Depth of field: The range of focal lengths for which a lens can keep the image in focus.

Diffusion length: In resistive networks and other physical systems based on diffusion equations, it is the exponential decay constant *tau* that expresses the influence of an input source on neighboring nodes.

Disparity: In stereoscopy it is the relative difference in retinal projections of the same point on the left and right eyes or cameras. The disparity is a direct measure of the object depth: a faraway object has a lower disparity than a closer one. See *depth*.

Disparity flat cells: Cells in the visual cortex that respond with equal strength to stimuli lying at any disparity.

Disparity gradient: In stereoscopy, the variation of perceived depth along the cyclopean angle.

EDA: Electronic Design Automation: refers to the set of CAD software tools used to automate or simplify the design and verification of integrated electronic circuits. These tools comprise schematic editors, mask layout editors, device simulators and verification tools.

EMD: Elementary Motion-Detectors: Each of the elementary cells capable of detecting motion by correlating a photodetector output with a delayed version of a neighboring photodetector output.

Epipolar constraint: A geometric constraint in stereoscopy that states that the points lying on a line on one eye or camera have their projections on the other eye or camera on the corresponding *epipolar line*. This constraint reduces the search space to find corresponding points. See *correspondence problem*.

Epipolar lines: In stereoscopy, for a given point in the scene, the epipolar lines are the projection on the right eye of the line connecting the point to the left-eye focus and the projection on the left eye of the line connecting the point to the right-eye focus. Thus, any point lying on one of the epipolar lines will find its corresponding projection on the other epipolar line. See *correspondence problem*.

Far cells: Cells in the visual cortex that respond maximally to binocular visual stimuli with low disparity, i.e., stimuli lying far away.

Feature extraction: In image processing it is the step whereby the most relevant points in a scene are identified. The relevance of a point depends on the particular application.

FET: Field-Effect Transistor: A type of transistor device based on the flow of current through a variable-width channel. It comes in two flavors: metal-oxide-semiconductor FET (MOSFET) (see *CMOS*) and junction FET (JFET).

Fixation point: In stereoscopy it is the point where both optic axes intersect. It is considered to be the point where the viewer's attention is focused.

Follower aggregation: An analog circuit that aggregates the output current of a number of voltage-followers to compute a weighted average. See *collective computation, voltage follower.*

Fovea: The high-resolution, central area of the retina.

FPGA: Field-Programmable Gate Array: A standard, commercially-available, digital VLSI circuit that contains many transistors (gates) whose interconnection can be electrically modified to generate any boolean logic function that can be produced with the available number of gates.

Fusion: In stereoscopy, it is the ability to create a single image, called cyclopean image, from a stereo pair of images.

Gábor function: A harmonic function with a Gaussian envelope, $g(x) = e^{-\frac{x^2}{2\sigma}} \cdot e^{j\omega_o x}$, with the key property of simultaneously minimizing the spatial and spatial-frequency uncertainties. Proposed by Dennis Gábor.

Gaussian filter: A lowpass filter with a Gaussian-shaped impulse response.

Gilbert multiplier: An analog circuit whose output is proportional to the product of its two differential voltage inputs. It is also referred to as four-quadrant multiplier because both inputs can be positive or negative. From B. Gilbert, who first proposed such circuit.

Horopter: In stereoscopy, it is the locus of zero disparity, that is, the surface whose points are projected on both eyes at the same retinal coordinates. See *Vieth-Müller circle.*

Hyperacuity: When a vision system has an acuity beyond the spatial resolution of its photodetectors, it is said to be endowed with hyperacuity. See *acuity.*

Implant optimization: A manufacturing procedure that has to be added to produce good-quality CMOS photosensors using standard CMOS processes. See *CMOS.*

Ipsilateral: In systems with a symmetric structure such as the human body, it refers to connections coming from the same side. For instance, ipsilateral connections bring some nerves of the left eye to the left hemisphere of the brain. Opposite: *contralateral.*

Laplacian of Gaussian: Refers to a filter whose impulse response has the shape of the Laplacian (second derivative) of a Gaussian function. This shape can be approximated by a difference of Gaussians (DOG).

Lateral geniculate nucleus (LGN): A six-layered nucleus in the brain (in the thalamus) that receives the optic fibers coming from the retinal ganglion cells. The LGN is retinotopically organized, with layers 1, 4 and 6 receiving inputs from the *contralateral* eye and layers 2, 3 and 5 from the *ipsilateral* eye.

Motion parallax: An apparent difference in the perceived motion depending on the target distance. Closer objects appear to move faster than faraway ones.

Near cells: Cells in the visual cortex that respond maximally to binocular visual stimuli with high disparity, i.e., stimuli lying nearby.

Negative impedance converter: A circuit based on a negative-gain amplifier that simulates "negative resistors" by producing a negative voltage to current ratio.

Occlusion: In stereoscopy, it is the blocking of one eye's line of sight by an object in the scene. Occlusions happen at object boundaries, and they are problematic because occluded areas can not be correctly matched. See *correspondence problem.*

Ocular dominance columns: Areas of the visual cortex that react mostly to stimuli from one eye, which is said to 'dominate'.

Optic chiasm: Point in the mammalian brain where the optic fibers coming from the eyes cross. See *decussation, contralateral, ipsilateral.*

Ordering constraint: In stereoscopy, it states that excluding occlusions, the ordering of points in the scene is conserved in the left and right eye projections.

Panum's fusional area: The range of disparities around the horopter where points can be correctly fused. Expressed in angle units. See *horopter, fusion.*

Photodiode: A class of photodetector based on a diode, i.e., a junction of donor-doped semiconductor with acceptor-doped semiconductor, that transforms the incoming light (photons) into current.

Phototransducer: A device capable of converting light intensity into electric magnitudes such as current or voltage. *Phototransistors, photodiodes* and *CCDs* are examples of such transducers.

Phototransistors: A class of photodetectors based on transistors, either bipolar or *CMOS.*

Pinch-off voltage: The voltage threshold between the *strong inversion* and *weak-inversion* zones of CMOS transistor operation.

Place coding: A coding technique whereby the value of interest is coded as the weighted average of a distribution of values.

Pseudoconductance: The conductance, that is the $\frac{I}{V}$ ratio, of a transistor seen as a two-port pseudodevice by considering only its drain and source for a given gate voltage.

Random-dot stereogram: A psychophysical experiment consisting in two images of randomly place dots where some of them are drawn at a relative shift (disparity). When trying to fuse both images, the dots will be perceived as lying at different depths. Invented by B. Julesz. See *fusion.*

Saturation: A CMOS transistor operating mode whereby the current flowing through it is independent of either the gate or drain voltage.

Standard cells: A set of circuits commonly used in electronic circuit that need not be redesigned and tested but rather can be taken from a standard-cell library and inserted in the current design.

Stereo correspondence: The processing stage of stereoscopy whereby for each retinal projection of a point in the scene its corresponding projection on the other retina is found.

Stereoacuity: The reciprocal of the minimum depth difference that can be resolved by a stereoscopic vision system. See *acuity*.

Stereopsis: The perception of depth through the processing of two or more images of the scene taken simultaneously and from different points of view. Also known as stereoscopy.

Strong inversion: A CMOS transistor operating region, also called *above threshold*. In this region the currents flowing into and out of the transistor follow a quadratic law.

Subthreshold region: A CMOS transistor operating region, also called *weak inversion*. In this region the currents flowing into and out of the transistor follow an exponential law.

Symbolic matching: In Computer Vision and Artificial Intelligence, it refers to the class of matching algorithms that operate on symbolic information as opposed to numerical information.

Symmetry constraint: In stereoscopy, it states that, if a point A on the left image corresponds to point B on the right image, point B should also correspond to point A.

Time-to-contact: For an object moving at a constant relative speed V to an obstacle, it the distance/speed ratio, which yields the time to collision.

Transconductance: In electronics, it is the $\frac{I_{out}}{V_{in}}$ ratio for a 4-port physical device.

Translinear element: In electronics, it is a device with a linear relationship between transconductance and current.

Translinear loop: In electronics, it is a closed loop containing an equal number of opposite connected translinear elements. See *translinear element*.

Tuned cells: Cells in the visual cortex that maximally respond to stimuli lying at a particular disparity.

Uncrossed disparity: In stereoscopy, when in order to fixate on a point the vergence has to be decreased, i.e., the optic axes need to be further uncrossed, the given point is said to be at an uncrossed disparity with respect to the current fixation point, that is farther away from the observer. See *fixation point, horopter*.

Uniqueness constraint: In stereoscopy, it states that a point on the left image can correspond with one and only one point on the right image.

V1: An area of the visual cortex through which most of the connections from the lateral geniculate nucleus (LGN) enter the cortex. It is also called striate cortex and primary visual cortex. See *cortex*.

Vergence angle: In stereoscopy, it is the angle at which the optic axes of the two eyes or cameras cross. See *fixation point*.

Vernier acuity: The reciprocal of the minimum angular separation between two points that can be resolved by a vision system. Often expressed in angle units or linear units on the focal plane.

VHDL: Very high-speed integrated circuit Hardware Description Language: A programming language that can be used to describe the behavior of digital circuit elements to help in validating circuit designs.

Vieth-Müller circle: The intersection of the horopter (a 3D surface) with the plane of the optic axes has a circular shape and it is called Vieth-Müller circle.

Visual cortex: An area of the brain, found on the outer layers of the occipital lobes, that is responsible for most of the visual processing in humans and upper vertebrates.

Voltage follower: An analog circuit, a kind of voltage amplifier with unity gain that is used to separate two circuit stages.

Weak inversion: A CMOS transistor operating region, also called *subthreshold* region. In this region the currents flowing into and out of the transistor follow an exponential law.

Winner-take-all: A multiple-input, analog, collective-computation circuit that identifies the input with the highest value. See *collective computation*.

Bibliography

[ADJ96] M. Adjouadi, F. Candocia, and Riley J. Exploiting Walsh-based attributes to stereo vision. *IEEE Transactions on Signal Processing*, 44(2):409–420, February 1996.

[AND90] T.L. Anderson and M. Donath. Animal behavior as a paradigm for developing robot autonomy. *Robotics and Autonomous Systems*, 6:145–168, 1990.

[AND91] A.G. Andreou, K.A. Boahen, et al. Current-mode subthreshold MOS circuits for analog VLSI neural systems. *IEEE Transactions on Neural Networks*, 2(2):205–213, March 1991.

[AND95] A. Andreou, R.C. Meitzler, et al. Analog VLSI neuromorphic image acquisition and pre-processing systems. *Neural Networks*, 8(7/8):1323–1347, 1995.

[API98] Olivier Michel. *Khepera API Reference Manual.* LAMI-EPFL, Lausanne, Switzerland, 1998.

[ARR97] X. Arreguit. Biologically-inspired low-level vision systems: VLSI implementation and applications. Tutorial at the ICANN'97, Lausanne, October 1997.

[ARR94] X. Arreguit and E. Vittoz. Perception systems implemented in analog VLSI for real-time applications. In *Proceedings of PerAC '94 - From Perception to Action.* IEEE Computer Society Press, 94.

[ARR96] X. Arreguit, A. van Schaik, et al. A CMOS motion detector system for pointing devices. In *IEEE International Solid-State Circuits Conference (ISSCC96)*, pages 98–99 + 426, 96.

[AYA89A] N. Ayache. *Artificial Vision for Mobile Robots. Stereo Vision and Multisensory Perception.* MIT Press, 1989.

[AYA89B] N. Ayache. *Vision stéréoscopique et perception multisensorielle.* Science informatique. InterEditions, Paris, 1989.

[AYA98] V. Ayala-Ramirez and M. Briot. Stabilisation d'image pour des applications en robotique mobile en environnement naturel. In *Actes des 9èmes Journées des Jeunes Chercheurs en Robotique (JJCR'9)*, pages 59–63, Clermont-Ferrand, France, June 1998. LASMEA.

[BAB95] Thomas Babu, B. Yegnanarayana, and Sukhendu Das. Stereo-correspondence using Gabor logons and neural networks. In *Proceedings of the International Conference on Image Processing (ICIP'95)*, pages 386–389, 1995.

[Bai91] W. Bair, Ch. Koch, et al. Computing motion vision using analog VLSI vision chips: an experimental comparison among four approaches. In *Proceedings of* 2^{nd} *international conference on microelectronics for NNs*, Munich, October 1991.

[Bas80] M.J. Bastiaans. Gabor's expansion of a signal into gaussian elementary signals. *Proceedings of IEEE*, 68(4):538–539, April 1980.

[Bea94] W. Beaudot. *Le traitement neuronal de l'information dans la rétine des vertébrés. Un creuset d'idées pour la vision artificielle.* PhD thesis, Institut National Polytechnique de Grenoble, France, 1994.

[Bea95] S.S. Beauchemin and J.L. Barron. The computation of optical flow. *ACM Computing surveys*, 27(3):433–467, September 1995.

[Bis96] G.M. Bisio, B. Crespi, L. Raffo, et al. A distributed adaptive architecture for analog stereo depth estimation. In *IEEE Int. Workshop on Neural Networks for Identification, Control, Robotics, and Signal/Image Processing (NICROSP'96)*, pages 360–367, Venice, August 1996.

[Bis97] G.M. Bisio, G.M. Bo, M. Confalone, et al. A current-mode computational engine for stereo disparity and early vision tasks. In *Proceedings of Microneuro '97*, pages 83–90, Dresden, September 1997.

[Bla90] C. Blakemore, editor. *Vision: coding and efficiency.* Cambridge University Press, 1990.

[Blo97] J.A. Bloom and T.R Reed. An uncertainty analysis of some real functions for image processing applications. In *Proceedings of the International Conference on Image Processing (ICIP'97)*, volume 3, pages 670–673, Santa Barbara, CA, 1997.

[Bro86] R. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotic Automation*, RA2(1):14–23, 1986.

[Bus92] P. Buser and M. Imbert. *Vision.* MIT Press, 1992.

[Car81] G.A. Carpenter and S. Grossberg. Adaptation and transmitter gating in vertebrate photoreceptors. *Journal of Theoretical Neurobiology*, 1:1–42, 1981.

[Cas93] G Casini, G. Fontanesi, and P. Bagnoli. *Vision, Brain, and Behavior in Birds*, chapter Binocular Processing in Frontal-Eyed Birds, pages 159–167. MIT Press, 1993.

[Cha98] N. Chauvin, G. Marti, and K Konolige. A contour method for real-time range image parsing. In *Proceedings of International Conference on Intelligent Robots and Systems (IROS'98)*, Victoria, Canada, October 1998.

[Chu92] P. Churchland and T.J. Sejnowski. *The computational brain.* MIT Press, 1992.

[CMO98] K. Diefendorff. CMOS image sensors challenge CCDs. *Microprocessor Report*, 12(8):7–15, June 1998.

[Del96] T. Delbrück and Mead. C.A. Analog VLSI phototrtansduction by continuous-time, adaptive, logarithmic photoreceptor circuits. CNS Memo 30, Caltech, Pasadena, CA 91125, April 1996.

[Der87] R. Deriche. Using Canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, 2(1):167–187, May 1987.

[Dew92] S.P. DeWeerth. Analog VLSI circuits for stimulus localization and centroid computation. *International Journal of Computer Vision*, 8:191–202, 1992.

[Die97] B. Dierickx. The human eye versus silicon. In *Proceedings of IEEE Workshop on CCD & AIS*, June 1997.

[Dio99] Ch. Diorio, B. Minch, and P. Hasler. Floating-gate MOS learning systems. In *Proceedings of the International Symposium on the Future of Intellectual Integrated Electronics (ISFIIE)*, pages 515–524, Sendai, Japan, 1999.

[Duc97] B. Duc. *Feature design: Applications to motion analysis and identity verification.* PhD thesis, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 1997.

[Fle93] D.J. Fleet and A.D. Jepson. Stability of phase information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1253–1268, December 1993.

[Fos98] E.R. Fossum. Digital camera on a chip. *IEEE Micro*, pages 8–15, May-June 1998.

[Fra92] N. Franceschini, J.M. Pichon, and C. Blanes. From insect vision to robot vision. *Phil. Trans. Royal Society London*, 337:283–294, 1992.

[Gab46] D. Gabor. Theory of communication. *Journal of the Institute of Electrical Engineers*, 93:429–459, 1946.

[Gam94] L.M. Gambardella and C. Versino. Robot Motion Planning Integrating Planning Strategies and Learning Methods. In *Proceedings of 2nd International Conference on AI Planning Systems*, 1994.

[Gra93] P.R. Gray and R.G. Meyer. *Analysis and Design of Analog Integrated Circuits.* John Wiley & Sons, 3 edition, 1993.

[Gra94] I.S Gradshteyn and I.M Ryzhik. *Tables of Integrals, Series and Products.* Academic Press, fifth edition, 1994.

[Har93] R.M. Haralick and L. Shapiro. *Computer and Robot Vision*, volume II. Addison-Wesley, 1993.

[Har98a] R. Harrison and Ch. Koch. A neuromorphic visual motion sensor for real-world robots. In *Workshop on Defining the Future of Biomorphic Robots. Proceeedings of IROS '98*, Victoria, Canada, October 1998.

[Har98b] R.R. Harrison and Ch. Koch. An analog VLSI model of the fly elementary motion detector. In M.I. Jordan, M.J. Kearns, and A. Solla, editors, *Proceeding of Advances in Neural Information Processing Systems 10*, pages 880–886. MIT Press, 1998.

[Hee88] D.J. Heeger. Optical flow using spatiotemporal filters. *International Journal of Computer Vision*, 2:279–302, 1988.

[Hen97a] R.D. Henkel. Constructing the cyclopean view. In *Proceedings of the International Conference on Artificial Neural Networks ICANN'97*, pages 907–912, October 1997.

[HEN97B] R.D. Henkel. Fast stereovision by coherence detection. `http://axon.physik.uni-bremen.de/~rdh/research/`, 1997.

[HOU89] D. House. *Depth perception in frogs and toads.* Springer-Verlag, 1989.

[HUB94] D. Hubel. *L'oeil, le cerveau et la vision.* Pour la science. Diffusion Belin, 1994.

[IME97] Imec scientific Report. Technical report, IMEC, Leuven, Belgium, 1997.

[IND97] G. Indiveri. Winner-take-all networks with lateral excitation. *Analog Integrated Circuits and Signal Processing*, 13(1/2):185–193, May/June 1997.

[JC87] J. Caesar. *De Bello Gallico.* Wolfgang Hering, Leipzig, 1987.

[JUL71] B. Julesz. *Foundations of Cyclopean Perception.* University of Chicago Press, 1971.

[KEM97] S. Kempainen. CMOS image sensors: Eclipsing CCDs in visual information? *EDN Europe*, pages 101–119, October 1997.

[KOB91] H. Kobayashi, J.L. White, and A.A. Abidi. An active resistor network for Gaussian filtering of images. *IEEE Journal of Solid-State Circuits*, 26:738–748, May 1991.

[KOC94] Ch. Koch, W. Bair, J.G Harris, T. Horiuchi, A. Hsu, and J. Luo. Real-time computer vision and robotics using analog VLSI circuits. In *Proceedings of the* 2^{nd} *Advances in Neural Information Processing Systems (NIPS2)*, pages 750–757, 1994.

[KON98] K. Konolige. Small vision systems: Hardware and implementation. In *8th International Symposium on Robotics Research*, Hayama, Japan, October 1998.

[KRA96] J. Kramer, R. Sarpeshkar, and Ch. Koch. Analog VLSI motion discontinuity detectors for image segmentation. *IEEE*, pages 620–623, 1996.

[KRA98] J. Kramer and G. Indiveri. Neuromorphic vision sensors and preprocessors in system applications. In *Proceedings of AFPAEC '98*, 1998.

[KRA99] J. Kramer. Künstliche Retina. *Bulletin de l'Association Suisse des Electriciens*, pages 19–25, March 1999.

[KRO86] E. Krotkov. Visual hyperacuity: representation and computation of high precision position information. *Computer Vision, Graphics, and Image Processing*, 33:99–115, 1986.

[LAN96] O. Landolt. Description et mise en œuvre du chip ED084V2A. Technical Report 16-11-95, Centre Suisse d'Electronique et Microtechnique, Neuchâtel, Switzerland, 1996.

[LAN98] O. Landolt. *Analog computation with place coding.* PhD thesis, Ecole Polytechnique Fédérale de Lausanne, Switzerland, 1998.

[LAZ86] J.P. Lazzaro. analog: A functional simulator for VLSI neural systems. Master's thesis, California Institute of Technology, 1986.

[LAZ88] J. Lazzaro, S. Ryckebusch, M. Mahowald, and C. Mead. Winner-take-all networks of O(N) complexity. Technical Report Caltech-CS-TR-21-88, Caltech, Pasadena, CA, 1988.

[LDM98] Y. López de Meneses and O. Michel. Vision sensors on the webots simulator. In *Virtual Worlds '98*, volume 1434 of *LNCS/AI*, pages 264–273, Paris, France, July 1998. Springer-Verlag.

[LSP98] J.-C. Bessaud, S. Vetsch, V. Messerli, et al. The Visible Human Project. In `http://visiblehuman.epfl.ch`, 1998.

[LUC81] T. Lucas, B.D.and Kanade. An iterative image registration technique with an application to stereo vision. In *Proceeedings of Imaging Understanding Workshop*, pages 121–130, 1981.

[MÄC99] P. Mächler. Personal communications, 1999.

[MAH89] M. Mahowald and T. Delbrück. *Analog VLSI Implementation of Neural Systems*, chapter Cooperative stereo matching using static and dynamic image features, pages 213–238. Kluwer Academic Publishers, Boston, 1989.

[MAH92] M. Mahowald. *VLSI analogs of neuronal visual processing: a synthesis of form and function.* PhD thesis, California Institute of Technology, May 1992.

[MAH94] M. Mahowald. *An analog VLSI System for Steroscopic Vision.* Kluwer Academic Publishers, 1994.

[MAR76] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194:283–287, October 1976.

[MAR82] D. Marr. *Vision.* W.H. Freeman and company, 1982.

[MAR93] Martin. *Vision, Brain, and Behavior in Birds*, chapter Producing the Image, pages 15–22. MIT Press, 1993.

[MAT90] Maja Mataric. A distrubuted model for mobile robot environment learning and navigation. Technical Report 1228, MIT AI Lab, 1990.

[MCF93] S.A. McFadden. *Vision, Brain, and Behavior in Birds*, chapter Constructing the Three-Dimensional Image, pages 47–61. MIT Press, 1993.

[MEA89] C. Mead. *Analog VLSI and Neural Systems.* Addison-Wesley, 1989.

[MEY98] G. Meynants, B. Dierickx, D. Scheffer, and Vlummens. J. A wide dynamic range CMOS stereo camera. In *Advanced Microsystems for Automotive Applications*, Berlin, March 1998.

[MIC98] O. Michel. Webots: Symbiosis between virtual and real mobile robots. In J.C. Heudin, editor, *Proceedings of the First International Conference on Virtual Worlds (VW'98)*, volume 1434 of *LNCS/AI*, Paris, france, July 1998. Springer-Verlag.

[MIL95] J.R. Millán. Reinforcement learning of goal-directed obstacle-avoiding reaction strategies in an autonomous mobile robot. *Robotics and Autonomous Systems*, 15:275–299, 1995.

[MIT90] G.J. Mitchison and G. Westheimer. *Vision: coding and efficiency*, chapter Viewing geometry and gradients of horizontal disparity, pages 302–309. Cambridge University Press, 1990.

[MOI96] A. Moini. Vision chips or seeing silicon. Technical report, Centre for GaAs VLSI Technologies, University of Adelaide, Australia, January 1996.

[MON94] F. Mondada, E. Franzi, and P. Ienne. Mobile robot miniaturization: A tool for investigation in control algorithms. In T. Yoshikawa and F. Miyazaki, editors, *Proceedings of the Third International Symposium on Experimental Robotics 1993*, pages 501–513. Springer Verlag, 1994.

[MOO91] A. Moore and Ch. Koch. A multiplication based analog motion detection chip. In B. Mathur and Ch. Koch, editors, *Visual Information Processing: From Neurons to Chips*, volume 1473, pages 66–75. SPIE, 1991.

[MOR94] A. Mortara and E.A. Vittoz. A communication architecture tailored for analog VLSI artificial neural networks: Intrinsic performance and limitations. *IEEE Transactions on Neural Networks*, 5(3):459–466, 1994.

[NAK85] K. Nakayama. Biological image motion processig: A review. *Vision Research*, 25(5):625–660, 1985.

[NAM95] W. Von Seelen et al. *A Neural Architecture for Autonomous Visually Guided Robots - results of the NAMOS Project*, chapter 3, pages 58–71. VDI Verlag, 1995.

[NOM96] Nomadic Technologies, Inc. *Nomad User's Manual*, 1996.

[OHT85] Y. Ohta and T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:139–154, March 1985.

[OPP89] A.V. Oppenheim and R.W. Schaffer. *Discrete-Time Signal Processing*. Prentice Hall, 1989.

[OVE88] I. Overington. *Computer Vision. A unified, biologically-inspired approach*. Elsevier, 1988?

[PET72] J.D. Pettigrew. The neurophysiology of binocular vision. *Scientific American*, 227:84–95, August 1972.

[PET90] J.D. Pettigrew. *Vision: coding and efficiency*, chapter Is there a single, most-efficient algorithm for stereopsis?, pages 283–290. Cambridge University Press, 1990.

[PIG98] Piguet. Notes de cours: Circuits complexes. EPFL, Lausanne, Switzerland, 1998.

[PÖL96] W. Pölzleitner and D. Cassasent. Stereo disparity computation using Gabor filters and feature selection. In *Proceedings of SPIE*, volume 2752, pages 268–280, 96.

[POG84] Poggio. *Dynamic Aspects of Neocortical Function*, chapter Processing of stereoscopic information in primate visual cortex, pages 613–635. John Wiley, 1984.

[PRA85] K. Prazdny. Detection of binocular disparities. *Biological cybernetics*, 52:93–99, 1985.

[Pro90] F. Rubio. *Procesado de señal. Notas de clase.* CPET-ETSETB, Universitat Politècnica de Catalunya, Barcelona, Spain, 1990.

[Qia97a] N. Qian. Binocular disparity and the perception of depth. *Neuron*, 18:359–368, 1997.

[Qia97b] N. Qian and Y. Zhu. Physiological computation of binocular disparity. *Vision Research*, 37:1811–1827, 1997.

[Raf95] L. Raffo. Adaptive resistive network for stereo depth estimation. *Eletronics Letters*, 31(22):1909–1910, October 1995.

[Raf96] L. Raffo. Analysis and synthesis of resistive networks for distributed visual elaborations. *IEE Electronics Letters*, 32(8):743–744, February 1996.

[Raf98] L. Raffo, S. Sabatini, G.M. Bo, and G. Bisio. Analog VLSI circuits as physical structures for perception in early visual tasks. *IEEE Transactions on Neural Networks*, 9(6):1483–1494, November 1998.

[Row96] M.D. Rowley and J.G. Harris. An edge enhancement technique for analog VLSI early vision applications. In *Proceedings of the International Conference of Neural Nettworks*, pages 874–879, June 1996.

[Rui95] Y. Ruichek and J-G Postaire. A neural implementation for high speed processing in linear stereo vision. In *Proceedings of the International Conference on Systems, Man and Cybernetics*, volume 5, pages 3902–3907, Vancouver, Canada, October 1995.

[Rui96] Y. Ruichek and J.-G. Postaire. A neural matching algorithm for 3-D reconstruction from stereo pairs of linear images. *Patter Recognition Letters*, 17:387–398, 1996.

[San88] T.D. Sanger. Stereo disparity computation using Gabor filters. *Biological Cybernetics*, 59:405–418, 1988.

[Sar96] R. Sarpeshkar, J. Kramer, G. Indiveri, and C. Koch. Analog VLSI architectures for motion processing: from fundamental limits to system applications. *Proceeding of the IEEE*, 84(7):969–987, July 1996.

[Sar98] R. Sarpeshkar. Analog versus digital: Extrapolating from electronics to neurobiology. *Neural Computation*, 10:1601–1638, 1998.

[Sav99] G. Savatic. Algorithme de vision stéréoscopique par filtres de Gabor. Rapport de projet de semestre, LAMI- Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, February 1999.

[SEx91] A. Saint-Exupéry. *Le petit prince.* Schoeningh, 1991.

[Sme98] F. Smeraldi, O. Carmona, and J. Bigün. Real-time head tracking by saccadic exploration and Gabor decomposition. In *Proceedings of the 5th International Workshop on Advanced Motion Control (AMC'98)*, pages 684–687, Coimbra, Portugal, July 1998.

[Sri96] J.S. Chahl and M.V. Srinivasan. Visual computation of egomotion using an image interpolation technique. *Biological Cybernetics*, 74:405–411, 1996.

[STE95] L. Steels. When are robots intelligent autonomous agents? *Robotics and Autonomous Systems*, 15(1-2):3–9, July 1995.

[STO98] A. Stocker and R. Douglas. Computation of smooth optical flow in a feedback connected analog network. In *Proceedings of Advances in Neural Information Processing Systems (NIPS11)*, December 1998.

[SVM] K. Konolige. SRI stereo engine and Small Vision Module. `http://www.ai.sri.com/~konolige/svm/`.

[TAN89] C. Mead. *Analog VLSI and Neural Systems*, chapter Optical Motion Sensor, pages 229–255. Addison-Wesley, 1989.

[TAR91] L. Tarassenko, M. Brownlow, et al. Real-time autonomous robot navigation using VLSI neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, volume 3, pages 422–428. Morgan Kaufmann, 1991.

[TEL98] S. Shamma, A. Cohen, T. Horiuchi, and G. Indiveri. Workshop on Neuromorphic Engineering. Report to the National Science Foundation, NSF, Telluride, CO, June-July 1998.

[TOM97] P. Toombs. *Reinforcement Learning of Visually Guided Spatial Goal Directed Movement*. PhD thesis, Psychology Department, University of Stirling., 1997.

[TOV96] M.J. Tovée. *An introduction to the Visual system*. Cambridge University Press, 1996.

[TRA98] R. Trapp, S. Drüe, and G. Hartmann. Stereo matching with implicit detection of occlusions. In H. Burkhardt and B. Neumann, editors, *Proceedings of the Fifth European Conference on Computer Vision*, volume 2, pages 17–33, Berlin, June 1998. Springer Verlag.

[VEN96] Ph. Venier, O. Landolt, P. Debergh, and X. Arreguit. Analog CMOS photosensitive array for solar illumination monitoring. In *Proceeding of the IEEE International Solid-State Circuits Conference*, pages 96–97, 1996.

[VEN97] Ph. Venier. A contrast sensitive silicon retina based on conductance modulation in a diffusion network. In *Proceedings of Sixth International Conference on Microelectronics for Neural Networks, Evolutionary and Fuzzy Systems. MicroNeuro'97*, pages 145–148, Dresden, September 1997.

[VER96] C. Versino and L.M. Gambardella. Ibots. Learning Real Team Solutions. In *Proceedings of the 1996 Workshop on Learning, Interaction and Organizations in Multiagent Environments*, 1996.

[VIT94] E. Vittoz. Analog VLSI signal processing: Why, where and how? *Journal of VLSI Signal Processing*, 8:27–44, 1994.

[VIT96] E.A. Vittoz and X. Arreguit. Tutorial: Systems based on bio-inspired analog VLSI. In *International Conference on Microelectronics for Neural Networks and Fuzzy Systems (MicroNeuro '96)*, Lausanne, February 1996.

[VIT97] E. Vittoz. Pseudo-resistive networks and their applications to analog computation. In *Proceedings of 6th Conference on Microelectronics for Neural Networks and Fuzzy Systems (MicroNeuro'97)*, Dresden, Germany, September 1997.

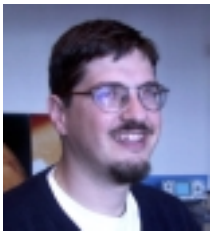
[WAL90] Jerry A. Waldvogel. The bird's eye view. *American Scientist*, 78:342–353, jul 1990.

[ZAN98] A. Zanela and S. Taraglio. Sensing the third dimension in stereo vision systems: a cellular neural networks approach. *Engineering Applications of Artificial Intelligence*, 11:203–213, 1998.

[ZEI93] H.P. Zeigler and H.-J. Bischof, editors. *Vision, Brain, and Behavior in Birds*. MIT Press, 1993.

Yuri LOPEZ DE MENESES
yuri@ieee.org

Born in August 24th, 1970
Spanish Nationality



Languages

Spanish:	Mother tongue.
English:	Fluent. Two 1-year stays in USA.
French:	Fluent. 4-year stay in France and Switzerland.
Catalan:	Fluent.
Russian:	Mother tongue.
German:	Read and spoken.
Italian:	Read and spoken.

Computing and technical skills

Languages:	C++, 80x86 and Motorola 68000 Assembler, Lisp, Prolog, Grafcet.
Operating systems:	UNIX (Solaris/Linux), VMS, Khepera OS.
Engineering tools:	Matlab, Concept, SPICE, ViewLogic, AnaLog.
Text editors:	Latex, HTML, FrameMaker.

Education

1996-1999	Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland, **Ph.D. in Computer Science.** November 1999.
1995	University of Southern California (USC), Los Angeles, USA, **Master's in Electrical Engineering.** Major in Robotics and Control.
1993-1994	Ecole Nationale Supérieure des Télécommunications de Bretagne (ENSTB), France, **Master's in Image Processing and Artificial Intelligence** .
1988-1994	Polytechnic University of Catalonia (UPC), Barcelona, Spain, **Telecommunications Engineer.** Major in Electronics.
1988	Barcelona, Spain, **Scientific Baccalaureate** .

Continuing Education

1997	EPFL, Lausanne, Graduate Course *"Computer Science and Biology: from the model to the tool"*, 200 hours.
1996	EPFL, Lausanne, Course *"UNIX Administration"*, 40 hours.
1993	UPC, Barcelona, Course *"Programmable Logic Controllers"*, 40 hours.

Experience

Feb. 1996 - Sept. 1999 Laboratoire de Microinformatique (LAMI), EPFL, Lausanne
Research Assistant

- Research and development of robotic vision systems based on the interfacing of analog sensors with microprocessors, in cooperation with the Centre Suisse d'Electronique et de Microtechnique (CSEM SA), Neuchâtel.
- Participation in the development and test of a vision sensor simulator for the Webots simulator, in cooperation with Cyberbotics SA.
- Administration of a 40-machine UNIX network. Management of the CHF 30,000.- annual budget for the lab's UNIX network.
- Creation of a hypertext document for on-line help to UNIX users.
- Organization of an international, French-speaking conference of Robotics Ph.D. students (`http://www.epfl.ch/jjcr11/`).
- Coordination of the Biovision and Robotics group.
- Supervision of semester and diploma projects at LAMI. Technical and system support for the microcontroller and computer systems labs.
- Reviewer for *IEEE Transactions on Circuits and Systems II.*

Sept.- Dec. 1995. Microprocessor Lab, University of Southern California
Lab Assistant

- Teaching to senior-year students 80x86 Assembler and the use of development tools.
- Supervision of the students' assembly and test of a single-board 8086 computer.

Jan. - Nov. 1994. Ecole Nationale Supérieure des Télécommunications de Bretagne, France
Research assistant

- Development of a neural network software, simplifying the use of self-organized maps.
- Modification of a self-organized network, improving its recognition rate, within the frame of a data analysis project.
- Development of a data fusion algorithm to improve satellite image classification.

Hobbies

Sports: Mountaineering (planned and led several long hikes), cross-country and alpine ski, sailing and team sports.

Reading: Political history books and biographies. Foreign languages.

Music: Opera.

Publications and Awards

[1] Y. López de Meneses and O. Carmona, "A stereo vision system for the Khepera", In 1^{st} *International Khepera Workshop*, Paderborn, Germany, December 1999.

[2] Y. López de Meneses, "La rétine EDI: Réalité et simulation", In *Actes des Journées des Jeunes Chercheurs en Robotique*, pages 35–39, Clermont-Ferrand, France, June 1998.

[3] Y. López de Meneses and O. Michel, " Vision Sensors on the Webots Simulator", In *Virtual Worlds '98*, LLNCS, pages 264–273, Paris, France, July 1998.

[4] O. Carmona and Y. López de Meneses, "Etude des mouvements oculaires: application à une rétine dynamique", In *Actes de Journées des Jeunes Chercheurs en Robotique, JJCR'8*, Nancy, France, January 1998. CRAN-LORIA.

[5] Y. López de Meneses, "*Vision-based Robot Control", Technical report, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, May 1998.*

[6] J. López de Meneses, B. Solaiman, R. Koffi, and M.C. Mouchot, "An Information Fusion Method for Multispectral Image Classification Postprocessing", In *International Geoscience And Remote Sensing Symposium, IGARSS'95*, volume 1, pages 99–101, Florence, Italy, July 1995.

[7] Y. López de Meneses, "Méthodes hybrides en segmentation d'images. Post-traitement d'images de classification par fusion multisources", Master's thesis, Ecole Nationale Supérieur des Télécommunications de Bretagne (ENSTB), France, September 1994.

Recipient: **La Caixa Foundation Fellowship**, for graduate studies in the USA, 1995

Recipient: **Erasmus Scholarship**, for student exchange within the European Union, 1993-1994

About this document

This report has been typeset on LaTeX on a Linux machine. The source occupied 330 Kbytes and compiled in 60 seconds on a 400 MHz PC to produce a 718 Kbytes DVI file. From this file, a 13 Mbyte PostScript output was generated using `dvips`.

Figures and diagrams have been drawn using `Xfig 3.2` and imported into the document as LaTeX pictures or sometimes as PostScript files since the LaTeX picture does not contain all geometric figures. Graphic plots have been obtained with `Gnuplot 3.5` and imported as LaTeX pictures.

Circuit schematics have been drawn on `xcircuit 2.0` and imported as PostScript into the sources. The schematics have been either drawn from scratch or imported as `AnaLog .lgf` files.

Images have been converted from JPEG to PostScript with `xv 3.10` and imported into the document as such.